SUMÁRIO

1 OBJETIVO

Esta norma estabelece os procedimentos a serem utilizados na análise de *software* para avaliação de modelo de instrumentos/sistemas de medição de energia elétrica.

2 CAMPO DE APLICAÇÃO

Esta norma aplica-se aos laboratórios nela acreditados e à Dimel/Disme/Sinst.

3 RESPONSABILIDADE

A responsabilidade pela aprovação, revisão e cancelamento desta norma é do Sinst.

4 DOCUMENTOS DE REFERÊNCIA

Portaria Inmetro nº 221, de 23 de maio de 2022	Aprova a regulamentação técnica metrológica consolidada para sistemas de medição ou medidores de energia elétrica ativa e/ou reativa, eletrônicos, monofásicos e polifásicos e sistemas de iluminação pública.
Portaria Inmetro nº 232 de 08/05/2012	Vocabulário Internacional de Metrologia: Conceitos Fundamentais e Gerais e Termos Associados (VIM) - 1ª. Edição Luso-brasileira (2012)
Portaria Inmetro nº 163 de 06/09/2005	Vocabulário Internacional de Termos de Metrologia Legal

(continua)

OIML D 31	*General requirements for software controlled measuring instruments – OIML, 2008*
OIML D 11	*General requirements for electronic measuring instruments – OIML, 2004*
WELMEC Software Guide 7.2 Issue 5	*Measuring Instruments Directive 2004/22/EC – WELMEC, March 2012*
Resolução Normativa ANEEL Nº 1.000, de 7/12/2021	Estabelece as Regras de Prestação do Serviço Público de Distribuição de Energia Elétrica
NIT-Sinst-004	Processo de avaliação de software no Sinst

5 DOCUMENTOS COMPLEMENTARES

NIT-Sinst-003	Organização da documentação para o processo de avaliação de *software*
NIT-Sinst-020	Protocolo de comunicação serial para verificação de integridade de *software* em instrumentos de medição
PLAN-Sinst-019	Planilha de Registro de Avaliação de *Software* para Avaliação de Modelo de Instrumentos/Sistemas de Medição de Energia Elétrica.
DOQ-Dimel-009	Orientações para redação de memorial descritivo de *software* para medidores de energia elétrica
SP 800-22	*A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*
SP 800-57 Part 1	*Recommendation for Key Management, Part 1: General*
FIPS PUB 186-3	*Digital Signature Standard (DSS)*

6 SIGLAS

As siglas das UP/UO do Inmetro podem ser acessadas em: http://www.inmetro.gov.br/inmetro/pdf/regimento-interno.pdf.

AC	Análise do Código-Fonte
AD	Avaliação da Documentação de *Software*
AM	Avaliação de Modelo
EF	Ensaio Funcional de *Software*
FIPS	*Federal Information Processing Standard*
JTag.	*Joint Test Action Group*
NATs	*Network Address Translation*
PC	*Personal Computer*
RAM	*Random Access Memory*
RBMLQ-I	Rede Brasileira de Metrologia Legal e Qualidade - Inmetro
RTM	Regulamento Técnico Metrológico
RTOS	*Real Time Operating System* (Sistema Operacional em Tempo Real)
SDMEE	Sistema Distribuído de Medição de Energia Elétrica

SHA256 *Secure Hash Algorithm* 2 (Algoritmo de *hash* seguro) com valor de *hash* de 256 bits
SP *Special Publication*
TLI Terminal de Leitura Individual
TPM *Total Productive Maintence*

7 TERMOS E DEFINIÇÕES

7.1 Arquivo binário

Arquivo de computador que não está em formato texto, oriundo da compilação de um código-fonte, que contém *software* legalmente relevante.

7.2 Assinatura digital

Resultado proveniente de processo algorítmico, que assegura autenticidade, integridade, não repúdio, e autoria de uma medição ou arquivo digital.

7.3 Atacante

Pessoa não autorizada que pretende obter informações ou o controle de um sistema digital.

7.4 *Buffer overflow*

Situação anômala aonde um programa escreve dados em um *buffer* e ultrapassa os limites definidos, acessando memória adjacente e provocando um comportamento não esperado.

7.5 Cadeia legalmente relevante

Todo *hardware* e *software* envolvido no processo de medição que consiste na aquisição, processamento e publicação dos dados de medição.

7.6 Carga de *software*

Processo de transferência automática de *software* para o instrumento de medição usando qualquer meio apropriado local ou remoto sem a necessidade de romper selagem principal.

7.7 Caso de teste

Uma especificação contendo o estado do sistema, um conjunto de entradas, um processo e saídas esperadas no intuito de validar o sistema para um determinado requisito.

7.8 *Checksum*

Exemplo de código utilizado para verificar a integridade de dados recebidos e/ou recuperados. Código utilizado para verificar a integridade de dados transmitidos.

7.9 Diagrama de tempo

Representação gráfica dos entes que compõe um sistema e suas interações numa escala de tempo.

7.10 Documentação de *software*

Conjunto de arquivos digitais a ser entregue a Dimel/Disme/Sinst, para serem analisados em um processo de análise dos requisitos de *software*. Sinônimo para pacote de entrega. Ver NIT-Sinst-003 item 7.15.

7.11 Estado

Estabelece o atual conjunto de condições do sistema. Em um determinado estado o sistema apresenta um comportamento, aguarda por um gatilho ou executa alguma ação.

7.12 Instrumento com computador tipo U

Um instrumento com computador de propósito geral, geralmente baseado em um PC, que não obedece a definição de computador tipo P.

7.13 Instrumento com computador tipo P

Um instrumento com computador tipo P embarcado é caracterizado por:

a) o *software* é construído exclusivamente para fins de medição. Adicionalmente funções para a proteção de *software* e dados, para transmissão de dados e para carga de *software* são consideradas construídas para o propósito de medição;
b) a interface do usuário é dedicada ao propósito de medição;
c) um sistema operacional (OS) ou subsistemas podem ser incluídos se apenas o *software* legalmente relevante possui comunicação externa; se não permite carregar ou alterar programas, parâmetros ou dados ou executar programas; se não permite alterar o ambiente da aplicação legalmente relevante; e inclui controle de acesso e não permite uma mudança subsequente na configuração de controle de acesso; e
d) o ambiente de *software* é invariável e não há meios internos ou externos para programar ou alterar o *software* em seu status incorporado, salvo quando os requisitos de carga de *software* são atendidos.

7.14 Interface de entrada de usuário

Interface de interação com o instrumento através de um meio físico, como teclado ou *touchscreen*.

7.15 Interface de *software* protetora

Interface entre o *software* legalmente relevante e o *software* legalmente não relevante.

7.16 Interrupção

Interrupção de *hardware* é um sinal de um dispositivo enviado ao processador com intuito de interromper o fluxo usual de instruções para que uma exceção de execução seja tratada.

7.17 Legalmente relevante

Todos os módulos de *software* (programas, sub-rotinas, objetos, etc.) que executam funções legalmente relevantes ou que contêm domínios de dados legalmente relevantes formam a parte de *software* legalmente relevante de um instrumento de medição. Mais especificamente, isso inclui todos os módulos de *software* que:

a) têm impacto no cálculo de uma unidade de medida legal;
b) contribuem para funções como: exibir, proteger e armazenar dados legalmente relevantes;
c) identificam os *softwares* legalmente relevantes; ou
d) executam carga de *software* legalmente relevante.

7.18 Memória de dados

Área de memória com permissão de leitura e escrita (RAM) utilizada para salvar dados.

7.19 Memória de programa

Área de memória onde está gravado o programa que está sendo executado.

7.20 Memorial descritivo

Documento que descreve detalhadamente as implementações tecnológicas voltadas para o atendimento dos requisitos de segurança de *hardware* e *software*.

7.21 Modo de configuração

Modo que permite alterar parâmetro legalmente relevante.

7.22 Não legalmente relevante

Todo *software/hardware*/dados presentes no instrumento que não são legalmente relevantes.

7.23 Parâmetro legalmente relevante

Parâmetro de um instrumento de medição ou de um de seus subconjuntos sujeito ao controle legal.

7.24 Primitiva criptográfica

Algoritmos criptográficos de baixo nível bem estabelecidos na literatura que são utilizados para construir protocolos criptográficos para sistemas de segurança de computadores.

7.25 Rede aberta

Uma rede de participantes arbitrários (dispositivos com funções arbitrárias). O número, identidade e/ou localização de um participante pode ser dinâmico e desconhecido para outros participantes.

7.26 Rede fechada

Uma rede de um número fixo de participantes com uma identidade conhecida, funcionalidade e localização.

7.27 Registro de alterações (ou registro de auditoria)

Conjunto de dados contendo o registro de quaisquer eventos e/ou alterações no instrumento que sejam legalmente relevantes e passíveis de influenciar suas características metrológicas.

7.28 Registro de carga de *software* (legalmente relevante)

Registro de auditoria que armazena os eventos relacionados às operações de carga de *software* legalmente relevante no instrumento.

7.29 Requerente

Pessoa jurídica, pública ou privada, nacional ou estrangeira, sediada no Brasil, que desenvolva atividades de produção, montagem, criação, construção, transformação, importação, exportação, distribuição ou comercialização de instrumentos e que requer a avaliação de modelo de instrumento.

7.30 Requisitos específicos

O *software* e o *hardware* legalmente relevantes que empregarem funcionalidades tecnológicas específicas devem obedecer aos requisitos específicos do RTM em questão.

7.31 Requisitos gerais

O *software* e o *hardware* considerados legalmente relevantes devem satisfazer à totalidade dos requisitos gerais do RTM em questão.

7.32 Selagem principal

Selagem do instrumento de medição (lacre) que demonstra que o instrumento estará apto a operar mediante a verificação por parte do Órgão da RBMLQ-I ou por entidade autorizada.

7.33 Sistema operacional

Uma coleção de *software* e elementos de *firmware* que controlam a execução de programas de computador e fornecem serviços como alocação de recursos de computadores, controle de tarefas, controle de entrada / saída e gerenciamento de arquivos em um computador.

7.34 Tarifa branca

É uma modalidade de tarifa horo-sazonal oferecida para unidades consumidoras atendidas em baixa tensão.

7.35 Tarifa horo-sazonal

Estrutura caracterizada pela aplicação de tarifas diferenciadas de consumo de energia elétrica e de demanda de potência de acordo com as horas de utilização do dia e dos períodos do ano, conforme especificação a seguir:

a) tarifa azul: modalidade estruturada para aplicação de tarifas diferenciadas de consumo de energia elétrica de acordo com as horas de utilização do dia e os períodos do ano, bem como de tarifas diferenciadas de demanda de potência de acordo com as horas de utilização do dia; e
b) tarifa verde: modalidade estruturada para aplicação de tarifas diferenciadas de consumo de energia elétrica de acordo com as horas de utilização do dia e os períodos do ano, bem como de uma única tarifa de demanda de potência.

7.36 Trilha de auditoria

Registro cronológico das atividades do sistema que permitem a reconstrução e o exame da sequência de eventos e / ou mudanças em um determinado evento.

7.37 Verificação de integridade

Processo que verifica que os dados/*software*/parâmetros não foram alterados durante o uso, reparo, manutenção, transferência ou armazenamento sem que haja autorização do Inmetro.

8 MÉTODOS DE ANÁLISE

8.1 A análise de *software* de instrumentos/sistemas de medição de energia elétrica para avaliação de modelo será baseada nas seguintes fontes de evidências:

a) documentação de *software*;
b) ensaios funcionais; e
c) código-fonte das partes legalmente relevantes.

8.1.1 Ao se iniciar a análise de *software* de instrumentos/sistemas de medição de energia elétrica, o técnico responsável deverá realizar o estudo preliminar dos memoriais descritivos, manual operacional e código-fonte, de forma a familiarizar-se com o instrumento/sistema de medição de energia elétrica.

8.1.2 Para cada *software* legalmente relevante do instrumento/sistema de medição, o memorial descritivo deve fornecer a informação de sua versão atual.

8.1.3 Caso seja necessário, o técnico responsável poderá requisitar entrevista com representante do requerente para obter esclarecimentos sobre o funcionamento do *software* e/ou *hardware* do instrumento/sistema de medição de energia elétrica e auxiliar na realização de ensaios funcionais.

8.1.4 O requerente deve fornecer todos os periféricos que se comunicam com o instrumento/sistema de medição de energia elétrica para realização dos ensaios.

8.2 Os métodos de análise de *software* para fins de avaliação de modelo são a seguir relacionados: avaliação da documentação de *software* (AD), análise do código-fonte (AC) e ensaios funcionais de *software* (EF).

Nota 1 – O ensaio funcional também aborda aspectos construtivos do instrumento.

Nota 2 – As evidências geradas por essa norma podem ser registradas na PLAN-Sinst-019 ou em documento próprio do sistema de gestão da qualidade do laboratório que cubra os requisitos do RTM anexo a Portaria Inmetro nº 221, de 23 de maio de 2022.

8.3 Avaliação da documentação de *software* (AD)

8.3.1 A avaliação da documentação de *software* tem como objetivo verificar se as soluções tecnológicas apresentadas no memorial descritivo garantem a conformidade com os requisitos do regulamento técnico anexo a Portaria Inmetro nº 221, de 23 de maio de 2022. Essa avaliação é feita por meio da análise da completude dos documentos, análise dos diagramas de blocos que compõem o sistema e suas interfaces, análise do esquemático do *hardware*, análise de vulnerabilidades e análise textual.

8.3.2 Completude dos documentos

Consiste na análise do pacote de entrega perante os requisitos da NIT-Sinst-003.

8.3.2.1 O memorial descritivo de *software* é o documento central do pacote de entrega. Nesse documento o requerente explicita como cumpriu os requisitos da Portaria Inmetro nº 221, de 23 de maio de 2022. O DOQ-Dimel-009 orienta a organização do memorial de *software*.

8.3.2.2 O técnico responsável deve verificar se os documentos fornecidos pelo requerente evidenciam o cumprimento dos requisitos da Portaria Inmetro nº 221, de 23 de maio de 2022, e se as soluções tecnológicas empregadas são adequadas para garantir a integridade e segurança da medição e do instrumento em si.

8.3.2.3 Documentação adicional pode ser requerida ao requerente caso a análise do memorial descritivo e demais documentos não fornecerem evidências adequadas do cumprimento dos requisitos da Portaria Inmetro nº 221, de 23 de maio de 2022.

8.3.3 Análise do diagrama de blocos que compõem o sistema e suas interfaces

8.3.3.1 Verificar se estão presentes na documentação a descrição das partes que tornam possível o funcionamento do sistema como um todo, a comunicação entre todas essas partes e a infraestrutura que suporta e gerencia essa comunicação.

8.3.3.2 Para cada bloco do sistema/instrumento de medição de energia elétrica, verificar se estão descritos na documentação os recursos que suportam a execução do *software* (memória e seu mapa, processador/micro controlador), os aspectos estáticos (arquitetura de *software*, ambiente de desenvolvimento) e os dinâmicos (fluxos de execução) do *software* e as funcionalidades específicas do bloco que contribuem para o funcionamento do sistema como um todo.

8.3.3.3 Verificar, para cada interface de comunicação envolvida na manipulação de dados, se estão descritos protocolos e algoritmos utilizados, a estrutura dos quadros transmitidos e a tecnologia empregada.

8.3.4 Análise do esquemático do *hardware*

A análise do esquemático do *hardware* visa compreender a interação de todos os blocos do sistema/instrumento de medição de energia elétrica através de suas interfaces. Verificar se todos os blocos, interfaces de comunicação e os fluxos de informação estão representados e claramente indicados no esquemático. Verificar a existência de outros dispositivos microcontrolados.

8.3.5 Análise de vulnerabilidades

Verificar se a arquitetura proposta não apresenta vulnerabilidades documentadas na literatura que possam ser exploradas por um atacante, caso existam as medidas implementadas para sua mitigação devem estar claramente informadas no memorial descritivo O sistema/instrumento de medição de energia elétrica estará em conformidade se não forem constatadas vulnerabilidades.

8.3.6 Análise textual

Avaliar a hierarquização do conteúdo da documentação apresentada pelo requerente buscando identificar se as soluções tecnológicas apresentadas atendem os requisitos do Regulamento Técnico Metrológico.

8.4 Análise de código-fonte (AC)

8.4.1 A análise do código-fonte comentado tem como objetivo verificar a coerência da implementação dos programas embarcados em relação à documentação técnica depositada, por meio da inspeção do código-fonte, análise do fluxo de dados, da análise do fluxo de controle, da análise da completude dos comandos, do rastreamento das variáveis relevantes e da análise de vulnerabilidades.

8.4.2 Inspeção do código-fonte.

Revisão do código-fonte na busca de uma determinada informação para averiguar o atendimento à um requisito do RTM ou dessa norma.

8.4.3 Análise do fluxo de dados.

Verificar se os intervalos de valores das variáveis do programa estão respeitando os limites das mesmas. Caso existam intervalos que não respeitem os limites, inspecionar o comportamento e se violam o funcionamento do sistema. O sistema/instrumento de medição de energia elétrica estará em conformidade com os requisitos do RTM se os intervalos que não respeitem os limites não tiverem qualquer efeito sobre as funções do equipamento.

8.4.4 Análise do fluxo de controle.

Verificar se o fluxo lógico do programa está de acordo com o aspecto dinâmico (fluxo de execução) especificado na descrição de cada bloco do sistema. Caso existam diferenças entre o fluxo lógico e o de execução, inspecionar o comportamento do sistema. O sistema/instrumento de medição de energia elétrica estará em conformidade com os requisitos do RTM se as diferenças não tiverem qualquer efeito sobre as funções do equipamento.

8.4.5 Análise da completude dos comandos

8.4.5.1 Inspecionar o código-fonte em busca de todos os comandos descritos na lista completa de comandos, verificando se os parâmetros e seus respectivos tamanhos são iguais aos constatados na documentação. O sistema/instrumento de medição de energia elétrica estará em conformidade se os comandos estiverem alinhados aos requisitos do RTM.

8.4.5.2 Inspecionar o código em busca de comandos não descritos. Caso existam, inspecionar o comportamento dos mesmos e se violam o funcionamento do sistema. O sistema/instrumento de medição de energia elétrica estará em conformidade se todos os comandos e seus efeitos sobre as funções do instrumento estiverem corretamente descritos na documentação. Não deve haver comandos não descritos.

8.4.6 Rastreamento das variáveis relevantes

8.4.6.1 Identificar as variáveis relevantes do sistema, bem como seus intervalos de valores. O sistema/instrumento de medição de energia elétrica estará em conformidade se o intervalo de valores para cada variável relevante for válido.

8.4.6.2 Realizar o rastreamento (*tracing*) dessas variáveis. O sistema/instrumento de medição de energia elétrica estará em conformidade se os procedimentos que manipulam as variáveis estiverem permitidos e se a implementação desses procedimentos esteja refletida no aspecto dinâmico (fluxo de execução) estabelecido na descrição de cada bloco do sistema.

8.4.7 Análise de vulnerabilidades

8.4.7.1 Realizar a análise de possíveis condições provenientes de erros de implementação das interfaces. O sistema/instrumento de medição de energia elétrica estará em conformidade se não forem constatados erros de implementação das interfaces, diminuindo as possibilidades de exploração por um atacante.

8.4.7.2 Realizar a análise da validação das entradas permitidas nas interfaces do sistema/instrumento de medição de energia elétrica a fim de reduzir as possibilidades de violação da integridade do sistema. A ferramenta a ser utilizada deve ser escolhida considerando as características específicas do sistema/instrumento de medição de energia elétrica, de modo a aumentar as chances de identificar vulnerabilidades. O sistema/instrumento de medição de energia elétrica estará em conformidade se as entradas permitidas nas interfaces forem válidas.

8.4.7.3 Realizar uma análise de *buffer overflows*, seja pela injeção remota de código malicioso ou pela interrupção de seu funcionamento, a fim de constatar sua inexistência. A ferramenta a ser utilizada para a realização de *buffer overflows* deve ser escolhida considerando as características específicas do sistema/instrumento de medição de energia elétrica, de modo a aumentar as chances de identificar vulnerabilidades. O sistema/instrumento de medição de energia elétrica estará em conformidade se não forem identificados *buffer overflows*.

8.5 Ensaio funcional de *software* (EF)

8.5.1 Consiste na análise do comportamento do *software* legalmente relevante do instrumento/sistema de medição de energia elétrica em situações de operação real.

8.5.2 O ensaio funcional de *software* deve ser aplicado, quando requerido pelo técnico responsável, para assegurar, ratificar ou respaldar o atendimento aos requisitos do RTM.

8.5.3 Os procedimentos específicos dos ensaios funcionais de *software* devem tomar por subsídio as informações contidas na documentação de *software* do instrumento/sistema e equipamentos para simulação de medição.

8.5.4 As características descritas nos memoriais descritivos e manual operacional podem ser verificadas em procedimentos práticos por meio da realização de ensaios funcionais de *software*.

8.5.5 Através do ensaio funcional de *software*, deve ser analisada a operação normal do instrumento. Todas as chaves ou teclas e combinações descritas devem ser empregadas e a reação do instrumento/sistema avaliada. Para interfaces gráficas de usuário, todos os menus e demais elementos gráficos devem ser ativados e avaliados.

9 REQUISITOS GERAIS

9.1 Características básicas do instrumento/sistema de medição de energia elétrica

9.1.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.1 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.1.2 Avaliar a completeza da documentação e necessidade de atendimento a requisitos específicos.

9.1.3 Avaliação da documentação para características básicas do instrumento

9.1.3.1 Verificar a completude dos documentos.

9.1.3.2 Analisar os diagramas de blocos do instrumento presentes na documentação.

9.1.3.3 Analisar o esquemático do *hardware*.

9.1.3.4 Conferir se o mapa de memória é apresentado na documentação.

9.1.4 Análise de código-fonte para características básicas do instrumento

9.1.4.1 Analisar a necessidade da entrega do código-fonte do instrumento em questão. Caso afirmativo, inspecionar o código-fonte e verificar sua completude.

Nota – Garantir a completude do código-fonte pode ser uma tarefa não trivial.

9.1.4.2 Na documentação fornecida pelo requerente recomenda-se constar uma descrição técnica de como gerar os executáveis embarcados no instrumento. Essa documentação deve ser seguida e os arquivos de compilação automática conferidos, como *makefile*.

9.1.5 Ensaios funcionais para características básicas do instrumento

9.1.5.1 Verificar se, ao ligar, o instrumento/sistema de medição se comporta de acordo com os procedimentos especificados na documentação fornecida pelo requerente, apresentando as informações esperadas, inclusive reagindo da maneira prevista.

9.1.5.2 Verificar se o funcionamento do instrumento/sistema de medição está de acordo com a descrição contida na documentação fornecida pelo requerente, inclusive em relação ao relógio, postos horários e registro de energia ativa e/ou reativa.

9.1.5.3 Verificar se a interface de entrada de usuário se comporta do modo especificado na documentação fornecida pelo requerente.

Nota – As verificações funcionais das características básicas estão normalmente associadas ao manual operacional do equipamento.

9.2 Identificação de *software*

9.2.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.2 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.2.2 Cada *software* embarcado no instrumento deve ter um identificador de versão de *software*, isto é, uma sequência de caracteres que identifica o *software* univocamente.

9.2.3 Avaliação da documentação para identificação de *software*

9.2.3.1 Avaliar textualmente se a documentação descreve como o identificador é construído, como é estruturado e como pode ser visualizado no equipamento.

9.2.3.2 Avaliar textualmente se a documentação explicita a versão atual de cada *software* embarcado.

9.2.3.2.1 No caso de instrumento com computador tipo U, avaliar textualmente se a documentação explicita um identificador para o sistema operacional.

Nota – Esse identificador pode ser, por exemplo, a versão da distribuição Linux ou o número do *Service Pack do Windows* 7.

9.2.3.2.2 No caso de instrumento com computador tipo U, avaliar textualmente se a documentação explicita um identificador para cada componente do sistema que tenha sido modificado ou adicionado, como por exemplo, *drivers* e bibliotecas.

9.2.3.2.3 No caso de instrumento com computador tipo P com sistema operacional integrado (*embedded operating system*), como por exemplo, um RTOS, avaliar textualmente se há um identificador de versão para o sistema operacional integrado.

9.2.3.3 No caso de um *software* oriundo de uma modificação de modelo, avaliar textualmente se a documentação apresenta o identificador de *softwares* alterados corretamente modificados segundo a regra de construção do identificador.

9.2.4 Análise de código-fonte para identificação de *software*

9.2.4.1 Rastrear as variáveis ou constantes relevantes para o identificador de *software*.

9.2.4.2 Analisar o fluxo de controle para a exibição do identificador *software*.

9.2.5 Ensaios funcionais para identificação de *software*

9.2.5.1 Seguir o procedimento descrito na documentação encaminhada pelo requerente e verificar se é possível acessar o identificador de versão de *software*.

9.2.5.1.1 Na ausência de interface, a identificação de *software* deve ser afixada sobre o instrumento.

9.2.5.2 Verificar se a estrutura do identificador de versão segue as regras definidas na documentação.

9.2.5.3 Verificar se a versão de cada *software* embarcado é a mesma que apresentada na documentação.

9.2.5.4 Verificar se cada identificador de cada software legalmente relevante está claramente apresentado e não pode ser confundido com qualquer outro identificador.

9.3 Integridade do *software*

9.3.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.3 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.3.2 O procedimento de verificação de integridade de *software* tem como objetivo atestar que o *software* instalado no instrumento em verificação corresponde exatamente à versão previamente aprovada para utilização no instrumento. Para fins de aprovação de *software*, considera-se que, quando da realização de procedimentos de verificação de integridade, a realização em laboratório equivale à realização em campo.

9.3.3 Avaliação da documentação para integridade de *software*

9.3.3.1 Avaliar textualmente a existência de procedimento descrito para verificação de integridade de toda memória de programa assim como o manual de operação da ferramenta de verificação de integridade.

9.3.3.2 No caso do uso de uma metodologia de desafio resposta, tal como discutida no Anexo A da NIT-Sinst-020, avaliar textualmente se a documentação explicita como os espaços não utilizados de memória são preenchidos com números aleatórios.

9.3.3.2.1 Caso o instrumento implemente uma metodologia de desafio resposta utilizando o protocolo de comunicação descrito na NIT-Sinst-020, não há necessidade de entregar uma ferramenta de verificação de integridade. Nesse caso a operação de verificação de integridade é realizada com o auxílio do Dispositivo de Verificação de Integridade de *Software*.

9.3.3.2.2 Avaliar textualmente se a documentação descreve a fonte dos números aleatórios utilizados. Se for empregado números pseudoaleatórios o algoritmo utilizado deve atender os requisitos da norma SP 800-22.

Nota – A documentação não precisa apresentar todos os testes propostos na SP 800-22 para o algoritmo que gera números pseudoaleatórios. Existem na literatura vários algoritmos que são bem-sucedidos nos testes estatísticos propostos na SP 800-22.

9.3.3.2.3 É desejável que as áreas de memórias para armazenamento de dados fiquem indisponíveis durante a verificação de integridade. Avaliar textualmente se a documentação descreve a disponibilidade das áreas de armazenamento durante a verificação de integridade. Por exemplo, caso o medidor possua memória de massa, essa deve estar indisponível (por exemplo, removida fisicamente) durante o procedimento de verificação de integridade por desafio-resposta.

9.3.3.2.4 Avaliar textualmente se a documentação descreve a organização da memória de programa para confirmar que não há áreas de memória duplicadas que permitam falsear a verificação de integridade por desafio-resposta.

9.3.3.3 No caso do uso de uma metodologia de verificação de integridade proposta pelo requerente. Avaliar textualmente se a documentação descreve os algoritmos e mecanismos de verificação de integridade implementados.

9.3.3.3.1 Avaliar a existência de vulnerabilidades na solução de verificação de integridade proposta e apure se não existe uma forma de um *software* malicioso residente falsear a verificação de integridade.

Nota – Não é raro que a alternativa a verificação de integridade por desafio e resposta seja uma comparação *bit-a-bit* do *software* aprovado com o *software* residente no instrumento. Uma metodologia baseada na comparação *bit-a-bit* requer uma plataforma confiável para ser executada.

9.3.3.4 No caso de uso de sistema operacional, avaliar textualmente se a metodologia proposta pelo requerente verifica a integridade dos arquivos que compõe o sistema operacional.

9.3.3.5 No caso de uso de sistema operacional, avalie textualmente se a solução proposta pelo requerente garante a integridade do *bootloader* do sistema operacional.

9.3.3.6 No caso de uso de sistema operacional, avaliar textualmente se a solução proposta pelo requerente garante a integridade do *firmware* da placa mãe responsável por inicializar o *hardware*.

Nota – Algumas soluções de computação confiável podem garantir a integridade do *firmware* da placa mãe, *bootloader* e do sistema operacional. Exemplos dessas tecnologias são: chip TPM, *TrustZone do Arm* ou *Plataform Trust Techonology* da Intel.

9.3.4 Análise de código-fonte para integridade de *software*

9.3.4.1 Analisar o fluxo de controle das rotinas implementadas no código-fonte que realizam a verificação de integridade.

9.3.4.2 Inspecionar no arquivo do *linker* (*linker map file*) se a organização de memória corresponde àquela apresentada na documentação.

9.3.5 Ensaios funcionais para integridade de *software*

9.3.5.1 Caso o instrumento não implemente o protocolo de comunicação da NIT-Sinst-020 o requerente deve fornecer uma ferramenta de verificação e integridade.

9.3.5.2 O requerente deve fornecer uma ferramenta e o respectivo procedimento de operação para carga de *firmware* no instrumento.

Nota 1 – Essa ferramenta e procedimento são necessários mesmo que a carga só ocorra em modo de fábrica, como por exemplo, fazendo uso de uma interface JTag.

Nota 2 – A ferramenta de verificação de integridade é parte do *software* legalmente relevante.

9.3.5.3 O requerente deve fornecer, além do *firmware* íntegro, um *firmware* não íntegro a ser carregado no instrumento.

9.3.5.4 Verificar se a ferramenta de verificação de integridade atesta como íntegro um *firmware* íntegro carregado no instrumento/sistema de medição.

9.3.5.4.1 Verificar diversas faixas de memórias nos testes de verificação de *integridade*, incluindo faixas que contenham os números aleatórios.

9.3.5.5 Verificar se a ferramenta de verificação de integridade atesta como não íntegro um *firmware* não íntegro carregado no instrumento/sistema de medição.

9.3.5.5.1 Verificar diversas faixas de memórias nos testes de verificação de integridade, incluindo faixas que contenham os números aleatórios. Inclua na verificação de integridade a região não integra do *firmware* modificado.

9.4 Exatidão dos algoritmos e funções de medição

9.4.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.4 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.4.2 A avaliação da exatidão dos algoritmos e funções de medição poderá ser realizada unicamente através de ensaios funcionais metrológicos, em uma etapa do processo de AM diferente da avaliação de *software*.

9.4.3 Avaliação da documentação para exatidão dos algoritmos e funções de medição

9.4.3.1 Avaliar textualmente se a documentação descreve os algoritmos e funções de medição incluindo o cálculo realizado, a exatidão e os arredondamentos dos resultados.

9.4.4 Análise de código-fonte para exatidão dos algoritmos e funções de medição

9.4.4.1 Analisar se o fluxo de controle das rotinas implementadas no código corresponde as rotinas de medição documentadas.

9.4.4.2 Analisar o fluxo de dados das variáveis utilizadas para armazenar dados de medição.

9.4.4.3 Inspecionar se as variáveis utilizadas para armazenar dados de medição possuem exatidão numérica adequada.

9.4.4.4 Rastrear as variáveis utilizadas para armazenar dados de medição.

9.4.5 Ensaios funcionais para exatidão dos algoritmos e funções de medição

9.4.5.1 Verificar a exatidão da medição de energia ativa e reativa em pontos não usuais.

9.4.5.2 Verificar a acumulação da energia nos registradores correspondentes aos postos horários corretos.

9.5 Influência da interface de entrada de dados

9.5.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.5 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.5.2 Para avaliação da influência da interface de dados, o requerente deve fornecer uma lista de todos os comandos que podem ser enviados via interfaces presentes no instrumento. Esses comandos não podem alterar de forma inadmissível o comportamento de *softwares* legalmente relevantes, de parâmetros legalmente relevantes ou de dados de medição.

9.5.3 Avaliação da documentação para influência da interface de entrada de dados

9.5.3.1 Avaliar textualmente se a documentação apresenta uma lista detalhando cada um dos comandos disponíveis. Essa lista deve explicitar os dados trafegados, o efeito dos comandos sobre o *software* legalmente relevante, sobre os parâmetros legalmente relevantes e os dados de medição.

9.5.3.1.1 Caso os comandos permitam alteração no *software* legalmente relevante ou sobre os parâmetros legalmente relevantes confirmar a existência de autenticação ou rompimento de selagem principal, conforme itens 9.7, 9.8 e 10.3 desta norma.

9.5.3.1.2 Caso os comandos trafeguem por rede de comunicação interna, atentar para os requisitos do item 10.2 desta norma.

9.5.3.2 Avaliar textualmente se a documentação descreve as proteções do canal de comunicação contra intrusões.

Nota – As rotinas responsáveis por tratar a entrada de dados não podem permitir uma alteração não documentada no estado do instrumento. Por exemplo, é aceito como solução uma interface que propaga os comandos documentados e rejeita todo comando desconhecido ou não permitido sem que isso cause impacto sobre o *software* legalmente relevante, parâmetros legalmente relevantes e dados de medição.

9.5.3.3 Avaliar textualmente se a documentação descreve os ensaios realizados para validar a completude dos comandos.

9.5.3.4 Avaliar textualmente se a documentação descreve os ensaios realizados para validar a funcionalidade de cada comando.

9.5.4 Análise de código-fonte para influência da interface de entrada de dados

9.5.4.1 Analisar a completude dos comandos da interface de entrada de dados.

Nota – Atenção para comandos não documentados e comandos documentados não implementados.

9.5.4.2 Analisar o fluxo de dados para cada comando.

9.5.4.3 Analisar o fluxo de controle para a entrada de cada comando.

9.5.4.4 Analisar vulnerabilidades no fluxo de controle dos comandos e nas funções que manipulam os dados de entrada.

9.5.5 Ensaios funcionais para influência da interface de entrada de dados

9.5.5.1 O requerente deve fornecer cabos, adaptadores, interfaces, *drivers* e qualquer outro recurso tecnológico necessário para o ensaio de influência da interface de entrada de dados.

9.5.5.2 Verificar se o efeito dos comandos de interface do(s) protocolo(s) de comunicação são aqueles descritos na documentação fornecida pelo requerente.

9.5.5.3 Verificar se existem comandos de interface não declarados utilizando o(s) protocolo(s) de comunicação possível(is).

9.5.5.3.1 Por exemplo, após uma atualização para corrigir uma não conformidade encontrada por comando não documentado, tentar enviar o comando não documentado para garantir que o *software* foi de fato atualizado. No caso de *softwares* que possuem o modo de *debug* definidos em tempo de compilação, utilizar o conhecimento do código-fonte para enviar comandos de *debug* para o instrumento em ensaio.

9.5.5.4 Verificar se é possível realizar intrusão não autorizada no medidor, sob ensaio, através de comandos de interface.

9.5.5.4.1 Verificar se é possível, através de intrusão não autorizada, prejudicar as funções legalmente relevantes do medidor.

9.6 Proteção contra mudanças acidentais/não intencionais

9.6.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.6 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.6.2 O(s) *software*(s) legalmente relevante(s), os parâmetros legalmente relevantes e os dados de medição devem ser protegidos contra modificações acidentais ou não intencionais.

9.6.3 Avaliação da documentação para proteção contra mudanças acidentais/não intencionais

9.6.3.1 Avaliar textualmente se a documentação descreve os testes contra influências físicas imprevisíveis.

Nota – Exemplo de solução aceitável: verificação de *checksum* de toda a memória de programa, incluindo a área de parâmetros legalmente relevantes, realizada periodicamente. Caso a verificação falhe o instrumento deve disparar um comportamento de falha.

9.6.3.2 Avaliar textualmente se a documentação demonstra que as funções de usuário requisitam confirmação quando dados são alterados ou suprimidos.

9.6.3.3 Avaliar textualmente se a os parâmetros legalmente relevantes passam por verificação de plausibilidade.

9.6.3.4 Avaliar textualmente se os dados de medição são protegidos contra mudanças acidentais.

9.6.4 Análise de código-fonte para proteção contra mudanças acidentais/não-intencionais

9.6.4.1 Analisar o fluxo de controle para proteção contra mudanças acidentais.

9.6.4.2 Inspecionar no código-fonte se toda memória de programa, memória de dados de medição e parâmetros legalmente relevantes são cobertos pela proteção contra influências físicas imprevisíveis.

9.6.5 Ensaios funcionais para proteção contra mudanças acidentais/não-intencionais

9.6.5.1 O requerente deve fornecer arquivos binários, assim como o procedimento e ferramentas necessárias para carrega-los no instrumento, para serem utilizados nos ensaios funcionais descritos nos itens 9.6.5.2 até 9.6.5.6 desta norma. Esses arquivos binários devem permitir a realização de ensaios que evidenciem a proteção contra mudanças acidentais.

9.6.5.2 Caso seja possível alterar ou apagar dados via interface de entrada de usuário, verificar se o instrumento requer do usuário confirmações nas alterações.

9.6.5.3 Se for possível alterar parâmetros legalmente relevantes, verificar o comportamento do instrumento quando os parâmetros são alterados para valores fora da faixa de plausibilidade.

9.6.5.4 Verificar se o medidor reconhece alterações não intencionais em áreas legalmente relevantes de sua arquitetura.

9.6.5.5 Verificar como o medidor reage a alterações não intencionais em áreas legalmente relevantes de sua arquitetura.

Nota – Os ensaios dos itens 9.6.5.4 e 9.6.5.5 desta norma fazem uso de arquivos binários fornecidos pelo requerente conforme descrito em 9.6.5.1 desta norma.

9.6.5.6 Caso seja utilizado algum *checksum* para garantir a proteção contra mudanças não intencionais, calcular os *checksums* e comparar com os valores nominais apresentados na documentação.

9.7 Proteção contra mudanças intencionais não autorizadas

9.7.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.7 do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.7.2 Avaliar se, através de alguma interface de comunicação (óptica, serial, rádio frequência, etc.), de posse do *software* de comunicação e configuração do usuário, é possível realizar intrusão ou modificações não autorizadas.

9.7.2.1 Caso seja permitido carga de *software*, atentar-se para o item 10.3 desta norma.

9.7.2.2 Caso seja permitido alteração de parâmetros legalmente relevantes por interface de comunicação, atentar-se para o item 9.8 desta norma.

Nota – As medidas de proteção adotadas podem ser mecânicas, eletrônicas e/ou criptográficas.

9.7.3 Avaliação da documentação para proteção contra mudanças intencionais não autorizadas

9.7.3.1 Avaliar textualmente se a documentação descreve as medidas adotadas para proteção do *software* e dos parâmetros legalmente relevantes.

9.7.3.2 Algumas arquiteturas possuem registradores que podem ser configurados para permitir alterações intencionais. Avaliar textualmente se a documentação descreve como estão configurados esses registradores.

9.7.3.3 Realizar uma análise de vulnerabilidades nas soluções documentadas para a proteção contra mudanças intencionais não autorizadas.

9.7.4 Análise de código-fonte para proteção contra mudanças intencionais não autorizadas

9.7.4.1 Inspecionar o código-fonte em busca das soluções apresentadas na documentação.

9.7.4.2 Na existência de mecanismos de teste (*debug*) que permitam mudanças não admissíveis, verificar que os mesmos não estão acessíveis na versão de produção no código-fonte.

Nota – Por exemplo, na linguagem C, os mecanismos de *debug* podem estar definidos dentro da diretiva de pré-processamento #ifdef. Verifique se o símbolo utilizado para teste, como por exemplo #ifdef DEBUG, não está definido em alguma parte do código ou no *makefile*.

9.7.4.3 Algumas arquiteturas possuem registradores para configuração. Esses podem permitir alterações intencionais da memória do instrumento. Inspecionar se os registradores estão corretamente configurados a fim de impedir alterações intencionais.

Nota – Por exemplo, a família de microcontroladores PIC32MX tem registradores de configuração, como o DEVCFG0 que configura a proteção contra escrita da memória.

9.7.4.4 Rastrear as variáveis relevantes associadas aos parâmetros legalmente relevantes**.**

9.7.4.5 Rastrear as variáveis relevantes associadas aos dados de medição**.**

9.7.4.6 Realizar uma análise de vulnerabilidades no código-fonte das rotinas associadas à proteção contra mudanças intencionais não autorizadas.

9.7.5 Ensaios funcionais para proteção contra mudanças intencionais não autorizadas

9.7.5.1 Verificar se há um método de controle de acesso ao medidor ou proteção física que impeça a alteração de características legalmente relevantes do *firmware* por pessoas não autorizadas.

9.7.5.2 Verificar se o medidor impede o acesso físico ao processador e à memória.

9.7.5.3 Verificar se o plano de selagem presente na documentação corresponde à selagem presente na amostra ensaiada.

9.7.5.4 Caso seja utilizado algum *checksum* para garantir a proteção contra mudanças intencionais, calcular os *checksums* e comparar com os valores nominais apresentados na documentação.

9.7.5.4.1 Caso seja utilizado *checksum*, recomenda-se um algoritmo com nível de segurança de pelo menos 112 *bits*, por exemplo, SHA256.

9.7.5.5 Caso exista no código-fonte mecanismos de teste (*debug*) que permitam mudanças não admissíveis verificar se eles não estão presentes na versão embarcada no instrumento.

Nota 1 – Por exemplo, pode-se supor que exista um comando de interface de dados que só deveria ser acessado no código compilado com a macro DEBUG. Tentar enviar esse comando ao instrumento e ver se o mesmo encontra-se habilitado.

Nota 2 – No caso de uma incoerência entre o código-fonte apresentado e o comportamento do instrumento, o técnico pode requisitar uma compilação assistida do código-fonte entregue pelo requerente. Uma outra forma de averiguar uma dissonância entre o código-fonte e o binário é comparar as *strings* presente no binário e no código-fonte, mantendo-se alerta para macros de pré-processamento como ___LINE__ e __FILE__ que podem inserir *strings* no código-fonte.

9.8 Proteção dos parâmetros

9.8.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.8 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.8.2 Os parâmetros que fixam as características legalmente relevantes do sistema/instrumento de medição de energia elétrica só podem ser modificados se autorizados.

9.8.3 Avaliação da documentação para proteção dos parâmetros

9.8.3.1 Avaliar textualmente a documentação a respeito da proteção dos parâmetros.

9.8.3.2 Avaliar textualmente a documentação a respeito da visualização dos parâmetros.

9.8.3.3 Avaliar textualmente se a documentação descreve o armazenamento em memória não volátil dos registros de alterações de parâmetros legalmente relevantes.

9.8.3.3.1 Avaliar textualmente se a documentação garante que cada registro de alteração de parâmetro legalmente relevante contém, ao menos, a identificação do parâmetro, o valor antes e depois da alteração e o instante da alteração (*timestamp*).

9.8.3.4 Caso os parâmetros possam ser alterados é necessário que o instrumento entre em modo de configuração. Avaliar textualmente se a documentação explicita o fato que para entrar em modo de configuração é necessário a ruptura de selagem principal ou autenticação.

9.8.3.4.1 Para medidores tarifa horo-sazonal (excluindo-se os medidores de tarifa branca), deve-se verificar:

a) uso de registro de alterações de parâmetros (log) legalmente relevantes e de carga de *software*, com pelo menos 100 registros;
b) autenticação compulsória padrão norma SP 800-57 part 1 para alteração de parâmetros legalmente relevantes e carga de *software*;
c) impressão de texto obrigatório no manual do instrumento: "O Inmetro recomenda, como boa prática de segurança para o cumprimento do item 2.8 do Regulamento Técnico Metrológico aprovado pela Portaria nº 221/2022, o uso de chave (senha) de autenticação individual para cada medidor de energia elétrica.";
d) implementação de comando de protocolo de comunicação que possibilite a alteração da chave (senha) de autenticação; e
e) implementação de comando de protocolo de comunicação que possibilite habilitar a expiração da chave (senha) de autenticação.

9.8.3.4.2 Para medidores de tarifa branca, são requisitos mínimos:

a) uso de registro de alterações de parâmetros (log) legalmente relevantes e de carga de *software*, com pelo menos 100 registros;
b) autenticação compulsória padrão definido pela norma SP 800-57 part 1 para alteração de parâmetros legalmente relevantes e carga de *software*;
c) impressão de texto obrigatório no manual do instrumento: "O Inmetro recomenda, como boa prática de segurança para o cumprimento do item 2.8 do Regulamento Técnico Metrológico aprovado pela Portaria nº 221/2022, o uso de chave (senha) de autenticação individual para cada medidor de energia elétrica.";
d) implementação de comando de protocolo de comunicação que possibilite a alteração da chave (senha) de autenticação; e
e) implementação de comando de protocolo de comunicação que possibilite habilitar a expiração da chave (senha) de autenticação.

9.8.4 Análise de código-fonte para proteção dos parâmetros

9.8.4.1 Inspecionar o código-fonte em busca das medidas descritas na documentação.

9.8.4.2 No caso de medidores tarifa branca e medidores tarifa horo-sazonal, certificar que os algoritmos de autenticação utilizado estão de acordo com o SP 800-57 part 1.

9.8.4.3 No caso de medidores tarifa branca e medidores tarifa horo-sazonal, certificar que foi implementado comando de comunicação que possibilite a alteração de chave (senha) e expiração de chave (senha) de autenticação.

9.8.4.4 Para os medidores tarifa branca e tarifa hora-sazonal, verifique se a memória não volátil disponível é suficiente para armazenar ao menos 100 registros de alterações de parâmetro e de carga de *software*.

9.8.5 Ensaios funcionais para proteção dos parâmetros

9.8.5.1 Caso exista um modo de configuração que permite alterar parâmetros legalmente relevantes, verificar se ao alternar entre o modo de configuração e retornar ao modo de operação os parâmetros estão protegidos.

9.8.5.2 Caso exista um modo de configuração que permite alterar parâmetros legalmente relevantes, alternar o instrumento para o modo de configuração e realizar uma alteração dos parâmetros legalmente relevantes. Retornar ao modo de operação. Verificar se ao retornar ao modo de operação o instrumento está utilizando os novos parâmetros.

9.8.5.3 No caso de medidores tarifa branca e medidores tarifa horo-sazonal, verificar o procedimento de autenticação utilizado, conforme descrito nos itens 9.8.3.4.1 e 9.8.3.4.2 desta norma.

9.8.5.4 No caso de medidores tarifa branca e medidores tarifa horo-sazonal, verificar o procedimento para alteração de chave, conforme descrito nos itens 9.8.3.4.1 e 9.8.3.4.2 desta norma.

9.8.5.5 Realizar alterações de parâmetros legalmente relevantes e verificar o registro de alterações de parâmetros.

9.9 Detecção de falha

9.9.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.9 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

9.9.2 O *software* deve detectar distúrbios no fluxo de seu processamento usual.

9.9.3 Avaliação da documentação para detecção de falha

9.9.3.1 Avaliar textualmente se a documentação descreve as possíveis falhas que o instrumento pode sofrer.

Nota – As possíveis falhas de um determinado instrumento estão intimamente ligadas à arquitetura do instrumento e poderão variar consideravelmente entre diferentes instrumentos.

9.9.3.2 Avaliar textualmente se as seguintes falhas referentes aos requisitos gerais do regulamento estão documentadas:

a) alteração não permitida (intencional e acidental) na memória aonde está localizado o *software* legalmente relevante;
b) alteração não permitida (intencional e acidental) na memória aonde estão localizados os parâmetros legalmente relevantes;
c) falha na gravação de dados de medição; e
d) violação de selagem eletrônica (*tamperproofing*).

9.9.3.3 Caso a arquitetura do instrumento empregue tecnologias que necessitem da avaliação de requisitos específicos, avaliar textualmente se as seguintes falhas referentes a esses requisitos estão documentadas:

a) falha na carga de *software; e*
b) falha na comunicação de rede.

9.9.4 Análise de código-fonte para detecção de falha

9.9.4.1 Analisar o fluxo de controle para as falhas documentadas.

9.9.5 Ensaios funcionais para detecção de falha

9.9.5.1 Para cada falha reproduzível na bancada, colocar o instrumento no estado da respectiva falha. Verificar se as reações ocorrem do modo descrito na documentação fornecida pelo requerente.

9.9.5.1.1 Se o instrumento entra em estado de falha, verificar se o processo de medição foi interrompido.

Nota – Dependendo da arquitetura do instrumento, é possível que alguns estados de falha não sejam atingíveis na bancada.

9.10 Validação de *software*

9.10.1 Avaliar se o instrumento/sistema atende os requisitos do item 2.10 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

Nota – Os casos de teste documentados pelo requerente são uteis para montagem experimental dos ensaios funcionais documentados nessa norma.

9.10.2 Avaliação da documentação para validação de *software*

9.10.2.1 Avaliar se os casos de testes utilizados para validação do *software* compreendem todo os requisitos pertinentes do regulamento técnico em questão.

9.10.2.2 Avaliar se a documentação dos casos de testes está de acordo com a Tabela 1.

Tabela 1 – Exemplo de registro de caso de teste

Item	Descrição
Título	Título do caso de teste.
Autor	Nome do responsável pela execução do teste.
Resumo	Contém uma descrição do caso de teste, descrevendo a finalidade ou o objetivo do teste e o escopo.
Pré-condições	Para cada condição de execução, descreve o estado obrigatório do sistema antes do início do teste.

(continua)

Item	Descrição
Entradas	Para cada condição de execução, enumera uma lista dos estímulos específicos a serem aplicados durante o teste. Em geral, eles são denominados entradas do teste e incluem os objetos ou os campos de interação e os valores de dados específicos inseridos durante a execução deste caso de teste.
Procedimento	Para a execução do teste, são as ações que o usuário deve fazer para que o sistema possa cumprir com o que será testado.
Resultados esperados	É o estado resultante ou as condições observáveis esperadas como resultado da execução do teste. Observe que isso pode incluir respostas positivas e negativas (como condições de erro e falhas).
Resultados Encontrados	É o resultado da execução do teste. Observe que isso inclui respostas positivas e negativas.
Evidência dos resultados encontrados	Conjunto de informações que evidencia o resultado descrito no item anterior, tais como: *printscreen* da tela do sistema contendo o resultado, registro fotográfico ou gravação de vídeo, arquivo de log do sistema, bloco de dados trafegado como resposta, etc.
Pós-condições	Para cada condição de execução, descreve o estado ao qual o sistema deverá retornar para permitir a execução de testes subsequentes. Relatar somente em casos excepcionais.

Fonte: Sinst

10 REQUISITOS ESPECÍFICOS

10.1 Separação das partes legalmente relevantes

10.1.1 Avaliar, se pertinente, se o instrumento/sistema atende os requisitos do item 3.3 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

10.1.2 Os requisitos do item 3.3 do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022 devem ser cumpridos com intuito de comprovar a separação inequívoca do *software* em *software* legalmente relevante e *software* legalmente não relevante. Se não existir separação de *software*, todo o *software* é considerado como legalmente relevante.

10.1.3 Avaliação da documentação para separação das partes legalmente relevantes

10.1.3.1 Avaliar se a documentação apresenta os seguintes itens:

10.1.3.1.1 Esquemático completo do sistema/instrumento de medição de energia elétrica apontando a(s) parte(s) legalmente relevante(s) e não legalmente relevante(s) de forma clara.

10.1.3.1.2 Descrição de todas as funções de programa e estruturas de dados legalmente relevantes.

10.1.3.1.2.1 Não deverá existir nenhuma função não documentada.

10.1.3.1.3 Descrição de todos os componentes que pertencem ao *software* legalmente relevante e sua inter-relação com as funções.

10.1.3.1.4 Descrição da interface de *software* protetora entre o *software* legalmente relevante e o *software* legalmente não relevante.

10.1.3.1.5 Lista completa de todos os comandos da interface de *software* protetora com atribuição inequívoca.

10.1.3.1.6 Uma declaração de completude, evidenciando que todos os comandos estão declarados na documentação.

Nota – Para a declaração de completude basta uma sentença clara na documentação que todos os comandos implementados no *software* estão descritos na documentação. Por exemplo: Declaro, para todos os fins, que todos os comandos presentes na interface do *software* legalmente relevante com o *software* legalmente não relevante são listados na tabela Z na seção X item Y.

10.1.3.1.7 Descrição dos comandos, dos dados trafegados e os seus efeitos sobre as funções e os dados do *software* legalmente relevante.

Nota – Caso a entrada de dados seja feita pelo *software* legalmente não relevante e esse se comunique com o *software* legalmente relevante ocorrerá uma superposição dos requisitos do item 10.1 e do item 9.5 desta norma. Entretanto, vale ressaltar que, apesar de comum, esse não é um *design* obrigatório. Um dado requerente pode ter *software* legalmente não relevante se comunicando com *software* legalmente relevante por um canal e a entrada de dados ser feita por outro canal.

10.1.3.2 No caso da existência de apresentação compartilhada no sistema/instrumento de medição de energia elétrica (entre o *software* legalmente relevante e o *software* legalmente não relevante) deve ser explicitamente descrito:

a) o conjunto de informações passível de apresentação;
b) como é feita a apresentação; e
c) o *software* que realiza a apresentação.

10.1.3.3 Avaliar se nenhum aspecto legalmente relevante foi implementado na parte não relevante.

10.1.3.4 No caso de carga de *software* legalmente não relevante a documentação deve explicitar claramente que a mudança não afeta a parte legalmente relevante.

10.1.3.5 Analisar o esquemático de *hardware* e compare com o diagrama de blocos que compõe o sistema e a interface de *software* protetora. Ambos devem evidenciar a separação da parte legalmente relevante da parte não relevante.

10.1.3.5.1 O *software* legalmente não relevante não pode alterar memória do *software* legalmente relevante.

10.1.3.6 Analisar eventuais vulnerabilidades na interface de *software* protetora.

10.1.4 Análise de código-fonte para separação das partes legalmente relevantes

10.1.4.1 Analisar a completude de comandos da interface de *software* protetora.

10.1.4.2 Inspecionar o código-fonte em busca de vulnerabilidades nas rotinas associadas a interface de *software* protetora.

10.1.4.3 Inspecionar o código-fonte em busca de aspecto legalmente relevante relegado ao *software* legalmente não relevante.

10.1.4.4 Analisar o fluxo de controle dos comandos permitidos pela interface de *software* protetora.

Nota – Nas arquiteturas com separação de *software* não é raro que o *software* legalmente não relevante disponha de maiores recursos de processamento ficando então responsável por realizar rotinas que necessitam maior poder computacional. Ao realizar a análise fique atento para garantir que uma mudança nessas rotinas não venha a impactar no processo de medição.

10.1.4.5 Analisar o fluxo de dados oriundo dos comandos permitidos pela interface de *software* protetora.

10.1.4.6 Inspecionar o código-fonte para garantir que todos os programas e bibliotecas envolvidos no processo de medição pertencem ao *software* legalmente relevante.

10.1.5 Ensaios funcionais para separação das partes legalmente relevantes

10.1.5.1 Verificar se existe uma correspondência biunívoca entre os esquemáticos que evidenciam a separação das partes legalmente relevantes e a organização dos componentes da amostra sob ensaio.

10.1.5.2 Para cada comando, verificar se o comportamento do instrumento corresponde ao documentado.

10.1.5.2.1 Investigar a existência de interações inadmissíveis.

10.1.5.2.2 Por exemplo, no caso de alguma não conformidade encontrada por comando não documentado, tentar enviar o comando não documentado para garantir que o *software* foi de fato atualizado. No caso de *softwares* que possuem o modo de *debug* definidos em tempo de compilação, utilizar o conhecimento do código-fonte para enviar comandos de *debug* para o instrumento em ensaio.

10.1.5.3 Caso a apresentação de informação seja compartilhada, verificar se a informação gerada pelo *software* legalmente relevante apresentada na saída do instrumento (por exemplo, *display*) pode ser identificada de forma inequívoca.

10.1.5.4 No caso de um instrumento com carga de *software*, verificar que a carga de *software* legalmente não relevante não altera o *software* legalmente relevante.

10.2 Transmissão de dados através de rede de comunicação

10.2.1 Avaliar, se pertinente, se o instrumento/sistema atende os requisitos do item 3.4 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

10.2.2 O conjunto de requisitos técnicos descritos a seguir, se aplica apenas quando o sistema/instrumento de medição de energia elétrica utiliza internamente à cadeia legalmente relevante de medição, uma rede de comunicação para transmitir e receber dados das medições.

10.2.3 Avaliação da documentação para transmissão de dados através de rede de comunicação

10.2.3.1 Analisar textualmente se a documentação descreve o protocolo de transmissão dos dados.

10.2.3.2 Analisar textualmente se todas as informações necessárias à apresentação, ou processamento, da medição são transmitidas.

10.2.3.3 Analisar textualmente se a documentação descreve a forma de verificar a integridade dos dados.

10.2.3.4 Analisar textualmente se a documentação descreve as ferramentas utilizadas para garantir a autenticidade dos dados transmitidos.

Nota – Não é raro o uso de selagem primária para garantir autenticidade em uma rede fechada. Essa abordagem é considerada correta, pois o rompimento de um lacre primário implica numa verificação metrológica e deixa uma trilha de auditoria. Entretanto, para uma rede aberta, o uso de selagem não garante a autenticidade da mensagem.

10.2.3.5 Caso o instrumento faça uso de ferramentas de criptografia para atender aos requisitos do item 3.4 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022, verificar por análise textual da documentação se os algoritmos apresentados estão de acordo com a SP 800-57 part 1.

10.2.3.5.1 Caso o instrumento faça uso de chaves criptográficas, verificar por análise textual da documentação as providências tomadas para que as chaves criptográficas utilizadas sejam mantidas em segredo, impossibilitando a extração das mesmas.

10.2.3.5.2 Caso o instrumento faça uso de chaves criptográficas, analisar textualmente as proteções contra mudanças não intencionais da chave criptográfica.

10.2.3.5.3 Caso o instrumento faça uso de chaves criptográficas, analisar textualmente as proteções contra mudanças intencionais da chave criptográfica.

Nota – Nos medidores do tipo SDMEE é recomendado, mas não é obrigatório, que cada TLI possua uma chave distinta para se comunicar com o medidor.

10.2.3.6 Analisar textualmente se a documentação descreve como os dados corrompidos são detectados e descartados.

10.2.3.7 Analisar textualmente se a documentação explicita a influência do tempo de comunicação. Esse não deve invalidar a medição, mesmo em alto tráfego de dados.

10.2.3.8 Caso os comandos trafeguem por rede de comunicação, avalie textualmente se a documentação descreve o método utilizado para garantir a integridade das mensagens.

10.2.3.9 Analise eventuais vulnerabilidades no uso das primitivas criptográficas.

Nota – Uma boa prática no uso de primitivas criptográficas é utilizar código que já foi amplamente testado. Verificar se a implementação das primitivas criptográficas utilizadas pertencem a bibliotecas que foram amplamente testadas.

10.2.4 Análise de código-fonte para transmissão de dados através de rede de comunicação

10.2.4.1 Analise o fluxo de controle que garante a integridade dos dados.

Nota – A integridade visa proteger os dados contra modificações não intencionais.

10.2.4.2 Analise o fluxo de controle que garante a autenticidade dos dados.

Nota – A autenticidade visa garantir a origem dos dados, isto é autenticidade da mensagem. Os mecanismos de autenticidade e integridade combinados visam proteger os dados contra mudanças intencionais.

10.2.4.3 Inspecione no código-fonte se os dados transmitidos só são utilizados após a sua integridade ser verificada.

10.2.4.4 Caso sejam utilizadas ferramentas de criptografia, verifique a se as ferramentas implementadas estão de acordo com a documentação.

10.2.4.5 Inspecione o código-fonte de forma a verificar se o pacote de dados transmitidos pela rede possui todas as informações necessárias à apresentação, ou processamento, da medição.

10.2.4.6 Realize uma análise de vulnerabilidade da comunicação entre os componentes do sistema de medição.

10.2.5 Ensaios funcionais para transmissão de dados através de rede de comunicação

10.2.5.1 Verificar se a comunicação entre os componentes do sistema de medição é realizada de acordo com o especificado na documentação fornecida pelo requerente.

Nota – Por exemplo, no uso de uma rede verificar portas abertas não documentadas, serviços de redes não documentados, NATs, etc.

10.2.5.2 Se aplicável, verificar se a troca de posição /ordem da conexão dos componentes do sistema de medição é detectada. Verificar se esta detecção provoca a reação descrita pelo requerente e se acarreta em registros errôneos de energia.

10.2.5.3 Verificar se a reação do sistema de medição a atrasos, interrupções e indisponibilidade de serviços é aquela descrita na documentação fornecida pelo requerente.

10.2.5.4 Indisponibilizar os serviços de rede de comunicação e verificar se não há perda dos dados de medição.

10.2.5.5 Verificar em ensaio se o usuário não é capaz de alterar os dados de medição suprimindo a transmissão dos dados.

10.3 Carga de *software* legalmente relevante

10.3.1 Avaliar, se pertinente, se o instrumento/sistema atende os requisitos do item 3.5 do Anexo B do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

10.3.2 Para instrumentos que permitam a carga de *software* em campo, se a assinatura digital for adotada pelo requerente como solução de autorização e autenticação, é necessário que o Inmetro realize procedimento de assinatura digital para validação da versão de *software* aprovada e teste-o antes da finalização do processo de avaliação de *software*.

10.3.3 Avaliação da documentação para carga de *software* legalmente relevante

10.3.3.1 Analisar textualmente se a documentação descreve o procedimento de carga de *software* e o uso de assinatura digital do Inmetro para autenticar o *software* legalmente relevante conforme descrito na NIT-Sinst-003 item 11.

10.3.3.1.1 Para cada algoritmo listado na FIPS 186 o Inmetro fornece uma chave de teste criptográfica para ser utilizada no processo de avaliação de *software*. O Anexo D da norma NIT-Sinst-003 apresenta uma chave teste para cada algoritmo aceito.

10.3.3.1.2 Para cada *software* daquele modelo que venha a passar no processo de avaliação de *software*, o requerente deve gerar uma assinatura digital de teste utilizando uma das chaves dispostas no Anexo D da NIT-Sinst-003.

10.3.3.1.3 A chave teste utilizada deve constar na documentação do requerente.

10.3.3.2 Analisar textualmente se a documentação descreve o comportamento do instrumento durante a carga de *software*.

10.3.3.2.1 O instrumento deve, no caso de uma carga de *software* malsucedida, retornar a versão de *software* previamente instalada ou evidenciar a falha com uma mensagem de erro permanente interrompendo seu funcionamento metrológico.

10.3.3.2.2 No caso de uma carga de *software* bem-sucedida, analisar textualmente se a documentação descreve como o instrumento restaura todas as proteções anteriormente ativadas, como é apagado o *software* antigo e se a área de memória restante é devidamente preenchida com dados aleatórios.

Nota – Por exemplo, ver se o instrumento que utilize algum tipo de senha retorna ao valor *default*.

10.3.3.3 Analisar textualmente se a documentação descreve como são armazenados em memória não volátil os registros de carga de *software*.

10.3.3.3.1 Analisar textualmente se a documentação garante que cada registro de carga de *software* contém, ao menos, a versão do *software* carregado (o antes e depois da carga) e o instante da carga (*timestamp*).

10.3.3.4 É necessário que o instrumento faça controle de permissão para realização da carga de *software* legalmente relevante. Verificar se a documentação explicita o mecanismo de permissão.

10.3.3.4.1 Para medidores tarifa horo-sazonal (excluindo-se os medidores de tarifa branca), deve-se verificar:

a) uso de registro de carga de *software* e de alterações de parâmetro, com pelo menos 100 registros;
b) autenticação compulsória padrão norma SP 800-57 part 1 para carga de *software*;

c) impressão de texto obrigatório no manual do instrumento: "O Inmetro recomenda, como boa prática de segurança para o cumprimento do item 2.8 do Regulamento Técnico Metrológico aprovado pela Portaria nº 221/2022, o uso de chave (senha) de autenticação individual para cada medidor de energia elétrica.";
d) implementação de comando de protocolo de comunicação que possibilite a alteração da chave (senha) de autenticação; e
e) implementação de comando de protocolo de comunicação que possibilite habilitar a expiração da chave (senha) de autenticação.

10.3.3.4.2 Para medidores de tarifa branca, são requisitos mínimos:

a) uso de registro de carga de *software* e de alterações de parâmetro, com pelo menos 100 registros;
b) autenticação compulsória padrão definido pela norma SP 800-57 part 1 para carga de *software* legalmente relevante;
c) impressão de texto obrigatório no manual do instrumento: "O Inmetro recomenda, como boa prática de segurança para o cumprimento do item 2.8 do Regulamento Técnico Metrológico aprovado pela Portaria nº 221/2022, o uso de chave (senha) de autenticação individual para cada medidor de energia elétrica.";
d) implementação de comando de protocolo de comunicação que possibilite a alteração da chave (senha) de autenticação; e
e) implementação de comando de protocolo de comunicação que possibilite habilitar a expiração da chave (senha) de autenticação.

10.3.4 Análise de código-fonte para carga de *software* legalmente relevante

10.3.4.1 Analisar o fluxo de controle das rotinas responsáveis por verificar a assinatura digital do *software* a ser carregado.

10.3.4.2 Analisar o fluxo de controle das rotinas responsáveis pelo controle de permissão para realização da carga de *software* legalmente relevante.

10.3.4.3 Analisar o fluxo de controle das rotinas que criam os registros de auditoria de carga de *software*.

10.3.4.4 Inspecione se as medidas descritas na documentação estão implementadas no código-fonte.

10.3.4.5 No caso de medidores tarifa branca e medidores tarifa horo-sazonal, certifique-se que os algoritmos de autenticação utilizados estão de acordo com o SP 800-57 part 1.

10.3.4.6 No caso de medidores tarifa branca e medidores tarifa horo-sazonal, certifique-se que foi implementado comando de comunicação que possibilite a alteração de chave (senha) e expiração de chave (senha) de autenticação.

10.3.4.7 Para os medidores tarifa branca e tarifa hora-sazonal, verifique se a memória não volátil disponível é suficiente para armazenar ao menos 100 registros de carga de software e de alterações de parâmetro.

10.3.5 Ensaios funcionais para carga de *software* legalmente relevante

10.3.5.1 Verificar se a carga autorizada de *software* é possível e se é realizada de acordo com a documentação fornecida pelo requerente.

Nota – Caso seja fornecida um *software* para realização de carga de *software*, esse é parte do *software* legalmente relevante.

10.3.5.2 Verificar se a carga não autorizada de *software* é negada e se a reação do instrumento é realizada de acordo com a documentação descrita pelo requerente.

10.3.5.3 Verificar se a reação à falha da carga *software* é realizada de acordo com a documentação descrita pelo requerente.

10.3.5.3.1 Verificar a reação à falha quando a carga de *software* é interrompida antes de sua finalização.

10.3.5.4 Verificar se o registro de auditoria para carga de *software* foi gravado corretamente e se pode ser examinado de acordo com a documentação descrita pelo requerente.

10.3.5.5 Verificar se os mecanismos de permissão para realização de carga de *software* foram implementados de acordo com a documentação e atendem os requisitos do item 10.3.3.4 desta norma e seus subitens.

10.4 Arquiteturas baseadas em assinatura digital

10.4.1 Avaliar, se pertinente, se o instrumento/sistema atende os requisitos do item 3.6 do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

10.4.2 O sistema/instrumento de medição poderá fazer uso de mecanismo de assinatura digital para assegurar a autenticidade e irrefutabilidade das informações de medição.

10.4.3 Avaliação da documentação para arquiteturas baseadas em assinatura digital

10.4.3.1 Avaliar textualmente se os algoritmos de assinatura digital documentados estão em acordo com a FIPS 186-3 e a SP 800-57 part 1.

10.4.3.2 Caso as grandezas de entrada sejam assinadas, analisar textualmente se documentação descreve a forma de reconstruir a medição a partir das grandezas de entrada.

10.4.3.3 Analisar eventuais vulnerabilidades no uso das primitivas criptográficas.

10.4.3.3.1 Uma boa prática no uso de primitivas criptográficas é utilizar código amplamente testado. Verificar se a implementação das primitivas criptográficas utilizadas pertencem a bibliotecas que foram amplamente testadas.

10.4.4 Análise de código-fonte para arquiteturas baseadas em assinatura digital

10.4.4.1 Analisar o fluxo de controle dos algoritmos de assinatura digital conforme explicitado na documentação.

10.4.5 Ensaios funcionais para arquiteturas baseadas em assinatura digital

10.4.5.1 Verificar se o algoritmo de assinatura digital corresponde àquele referenciado na documentação fornecida pelo requerente.

10.4.5.2 Verificar como é possível validar a autenticidade dos dados de medição através da assinatura digital, utilizando o algoritmo de criptografia adequado.

11 COMPORTAMENTO DINÂMICO

11.1 Avaliar se o instrumento/sistema atende os requisitos do item 3.7 do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

11.2 Em adicional aos requisitos de separação de *software*, descritos no item 10.1 desta norma, a coexistência de *software* não legalmente relevante não pode influenciar negativamente no comportamento dinâmico do processo de medição.

11.3 Avaliação da documentação para comportamento dinâmico

11.3.1 Analisar textualmente se documentação descreve como o *software* legalmente relevante tem prioridade no uso de recursos compartilhados com o *software* legalmente não relevante.

11.3.2 A documentação deve descrever a hierarquia das interrupções.

11.3.3 A documentação deve fornecer um diagrama de tempo das tarefas legalmente e das tarefas não legalmente relevantes.

11.3.4 A documentação deve apresentar casos de teste com as informações de desempenho do instrumento.

11.4 Análise de código-fonte para comportamento dinâmico

11.4.1 Verificar o código-fonte para validar a documentação a respeito da hierarquia de interrupções.

11.4.2 No caso do uso de um sistema operacional, verificar se os arquivos de configuração estão de acordo a garantir a prioridade do *software* legalmente relevante.

11.4.2.1 Por exemplo, num sistema utilizando Linux, verificar os valores de *priority* e *nice* do *software* legalmente relevante.

11.5 Ensaios funcionais para comportamento dinâmico

11.5.1 Verificar se o desempenho de processos legalmente relevantes do processador do medidor é influenciado (diminuído) pela realização de processos não legalmente relevantes. Disparar o máximo número de processos não legalmente relevantes, ao mesmo tempo em que se obtém o erro de medição de energia.

12 CAPACIDADE DE PROCESSAMENTO

12.1 Avaliar se o instrumento/sistema atende os requisitos do item 3.8 do RTM aprovado pela Portaria Inmetro nº 221, de 23 de maio de 2022.

12.2 O requerente deve apresentar todos os elementos constituintes do sistema/instrumento de medição de energia elétrica que tenham uso compartilhado.

12.3 Avaliação da documentação para capacidade de processamento

12.3.1 Analisar textualmente se a documentação apresenta os cálculos que comprovam a capacidade de processamento dos elementos constituintes que tenham uso compartilhado.

12.3.2 A documentação deve apresentar casos de teste com as informações de capacidade de processamento do instrumento.

12.4 Ensaios funcionais para capacidade de processamento

12.4.1 Se aplicável, verificar se os componentes de comunicação de uso compartilhado do sistema de medição foram dimensionados para os instantes de maior carga. Disparar o maior número possível de processos simultâneos de comunicação e verificar se a rede de comunicação utilizada é capaz de transmitir sem perda de desempenho.

13 HISTÓRICO DA REVISÃO E QUADRO DE APROVAÇÃO

Revisão	Data	Itens revisados
03	Dez/2022	▪ Atualização dos requisitos da Portaria Inmetro nº 21, de 23/05/2022; e ▪ Adequação a NIG-Gabin-040, Rev02.

Quadro de aprovação		
	Nome	**Atribuição**
Elaborado por:	Juliana Wilm Guedes	Auxiliar Administrativo
	Rogerio Possidonio Nunes	Pesquisador Tecnologista em Metrologia e Qualidade
Verificado por:	Alexandre Arosa Saturnino de Oliveira	Técnico em Metrologia e Qualidade
Aprovado por:	Ícaro dos Santos França	Chefe substituto do Sinst