## SUMÁRIO



## 1 OBJETIVO

A presente norma estabelece método a ser utilizado na avaliação de *software* para finalidade de avaliação de modelo de instrumentos de medição de velocidade automotivos.

## 2 CAMPO DE APLICAÇÃO

A presente norma aplica-se ao Sinst e a laboratórios acreditados.

## 3 RESPONSABILIDADE

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

## 4 DOCUMENTOS DE REFERÊNCIA

| | |
|---|---|
| Portaria Inmetro nº 232 de 08 de maio de 2012. | Adota, no Brasil, a 1ªedição luso-brasileira do Vocabulário Internacional de Metrologia – Conceitos fundamentais e gerais e termos associados (VIM 2012). |
| Portaria Inmetro nº 150 de 29 de março de 2016. | Adota, no Brasil, o Vocabulário Internacional de Termos de Metrologia Legal (VIML 2013). |
| OIML D 31 | *General requirements for software controlled measuring instruments – OIML, 2008* |
| *WELMEC Software Guide 7.2 Issue 5* | *Measuring Instruments Directive 2004/22/EC – WELMEC, March 2012* |
| Portaria Inmetro nº 158, de 31 de março de 2022. | Aprova o Regulamento Técnico Metrológico consolidado para medidores de velocidade de veículos automotores. |
| 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 | Requisitos do protocolo de comunicação serial para verificação de integridade de *software* em instrumentos de medição |
| 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 186-4 | *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 de código-fonte |
| AD | Análise da documentação de *software* |
| CPU | *Central Processing Unit* |
| EF | Ensaios funcionais de *software* |
| e.g. | *exempli gratia* (por exemplo) |
| GPS | *Global Positioning System* |
| ICP | Infraestrutura de chave pública |
| NAT | *Network Address Translation* |
| NTP | *Network Time Protocol* |
| RAM | *Ramdom Access Memory* |
| RBMLQ-I | Rede Brasileira de Metrologia Legal e Qualidade - Inmetro |
| RTM | Regulamento Técnico Metrológico |
| UTC | *Coordinated Universal Time* |

## 7 TERMOS E DEFINIÇÕES

### 7.1 Arquivo binário

Arquivo de computador em formato ilegível por humanos, 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 Ataque de força bruta

Método de obter autorização por tentativa e erro.

### 7.4 Autenticação

Verificação da identidade declarada ou alegada de um usuário, processo, ou instrumento de medição.

### 7.5 Autorização

Permissão concedida a um usuário ou processo para acessar um determinado recurso do sistema.

### 7.6 *Buffer overflow*

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

### 7.7 Cadeia legalmente relevante

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

### 7.8 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.

### 7.9 Caso de teste

Especificação contendo o estado do instrumento, um conjunto de entradas, um processo e resultados esperados no intuito de validar o cumprimento de um determinado requisito.

### 7.10 Certificado digital

Um conjunto de dados que identifica exclusivamente um instrumento de medição, contém a chave pública do instrumento e possivelmente outras informações, e é assinado digitalmente por uma parte confiável, vinculando assim a chave pública à entidade. Informações adicionais no certificado podem especificar como a chave é usada e seu período de validade. Ao decorrer do texto será utilizado apenas certificado.

### 7.11 *Checksum*

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

### 7.12 Diagrama de atividade

Fluxograma que demonstra a lógica de funcionamento de um determinado processo. Cada passo do processo, ou atividade, é denotado pela figura de uma caixa retangular e cada decisão denotada pela figura de um losango.

### 7.13 Diagrama de tempo

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

### 7.14 Documentação de *software*

Conjunto de arquivos digitais a ser entregue à Dimel/Disme/Sinst ou a quem ela delegar, para serem analisados em um processo de análise dos requisitos de *software*. Sinônimo para pacote de entrega.

### 7.15 Domínio de dados

Local e espaço na memória que cada programa necessita para processar dados.

Nota – Domínios de dados podem pertencer a somente um único módulo de *software* ou a vários.

### 7.16 Instrumento de medição construído com propósito específico (tipo P)

Instrumento de medição projetado e construído especialmente para a tarefa de medição.

### 7.17 Instrumento de medição universal (tipo U)

Instrumento de medição que compreende um sistema computacional contendo ao menos um computador de uso geral, para executar funções legalmente relevantes. Um instrumento de medição tipo U costuma ter os sensores externos a unidade principal de computação.

### 7.18 Interface de entrada de usuário

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

### 7.19 Interface protetora de *software*

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

### 7.20 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.

### 7.21 Legalmente relevante

Todos os módulos de *software* (programas, sub-rotinas, objetos, etc.) [sujeito ao controle legal] 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; e

**d)** executam carga de *software* legalmente relevante.

### 7.22 Memória de dados

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

### 7.23 Memória de programa

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

### 7.24 Memorial descritivo

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

### 7.25 Modo de configuração

Estado do instrumento de medição que permite alterar parâmetro legalmente relevante.

### 7.26 Parâmetro legalmente relevante

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

### 7.27 Primitiva criptográfica

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

### 7.28 Rede aberta

Uma rede de participantes arbitrários. O número, identidade e localização de um participante podem ser dinâmicos e desconhecidos para outros participantes.

### 7.29 Rede fechada

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

### 7.30 Requerente

Pessoa jurídica (ou seu representante legal), 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.

### 7.31 Registro de alterações/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.

### 7.32 Selagem primária

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 pelo Inmetro.

### 7.33 Selagem secundária

Selagem do instrumento de medição (lacre) que demonstra que o instrumento estará apto a operar mediante a verificação por parte do requerente, mediante controle de sua carga numérica aplicada.

### 7.34 Sistema operacional

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

### 7.35 *Software*/*hardware* de prateleira

Sistema de computador produzido de maneira uniforme, não customizável, e colocado no mercado para aquisição por qualquer parte interessada.

### 7.36 Trilha de auditoria

Conjunto de dados contendo os registros de auditoria dispostos em ordem cronológica de quaisquer eventos e/ou alterações no instrumento que sejam legalmente relevantes e passíveis de influenciar suas características metrológicas.

### 7.37 Verificação de integridade

Processo que verifica que os dados/*software*/parâmetros legalmente relevantes 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 de medição de velocidade de veículos automotores, doravante denominados apenas instrumentos, para fins de avaliação de modelo será baseada nas seguintes fontes de evidências:

**a)** documentação do processo de avaliação de *software,* conforme descrito na NIT-Sinst-003;
**b)** código-fonte das partes legalmente relevantes; e
**c)** ensaios funcionais.

**8.1.1** Ao iniciar a análise de *software* do instrumento, o técnico responsável deve realizar o estudo preliminar da documentação de forma a familiarizar-se com ele.

**8.1.2** Para cada *software* legalmente relevante do instrumento, a documentação de *software* deve fornecer a informação de sua versão em aprovação.

**8.1.3** Caso seja necessário, o técnico responsável poderá requisitar entrevista com o requerente ou seu representante devidamente autorizado para obter esclarecimentos adicionais sobre o funcionamento do *software* e/ou *hardware* do instrumento e auxiliar na realização dos ensaios funcionais.

**8.1.4** O requerente deve fornecer todos os periféricos e ferramentas que se comunicam com o instrumento para realização dos ensaios funcionais.

**8.2** As fases de análise de *software* para fins de avaliação de modelo são a seguir relacionadas:

**a)** avaliação da documentação de *software* (AD);
**b)** análise do código-fonte (AC); e
**c)** ensaios funcionais de *software* (EF).

### 8.3 Avaliação da documentação de *software* (AD)

**8.3.1** A completude dos documentos consiste na análise do pacote de entrega perante os requisitos da NIT-Sinst-003. Na evidência da falta de documentação, esta deve ser relatada e a análise interrompida. São itens obrigatórios, entre outros, o código-fonte, memorial descritivo do *software* e *hardware* e todos os itens da documentação requerida no RTM.

**8.3.1.1** O memorial descritivo de *software* e *hardware* faz parte do pacote de documentos apresentados na solicitação de avaliação de *software*. Nesse documento, o requerente explicita como cumpriu os requisitos do RTM. A norma NIT-Sinst-003 orienta a organização do memorial de *software* e *hardware*.

**8.3.1.2** Os itens do memorial descritivo devem ser apresentados na mesma sequência em que os requisitos são elencados no RTM.

**8.3.2** 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 RTM. Essa avaliação é feita por meio da análise da completude dos documentos, análise dos diagramas de blocos que compõem o instrumento e suas interfaces, análise do esquemático do *hardware*, análise de vulnerabilidades e análise textual.

**8.3.2.1** O técnico responsável deve verificar se os documentos fornecidos pelo requerente evidenciam o cumprimento dos requisitos do RTM, e se as soluções empregadas são adequadas para garantir a integridade, autenticidade e segurança da medição do instrumento.

**8.3.2.2** Documentação adicional pode ser solicitada ao requerente caso a análise do memorial descritivo e demais documentos não forneçam evidências adequadas do cumprimento dos requisitos do RTM.

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

**8.3.3.1** Verificar se estão presentes, na documentação, a descrição das partes que tornam possível o funcionamento do instrumento 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 que compõe o instrumento, verificar se estão descritos na documentação os recursos que suportam a execução do *software* (memória e seu mapa, processador/microcontrolador), 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 instrumento 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*

**8.3.4.1** A análise do esquemático do *hardware* visa compreender a interação de todos os blocos do instrumento 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

**8.3.5.1** Verificar se a arquitetura proposta não apresenta vulnerabilidades documentadas na literatura que possam ser exploradas e, caso existam, se os meios de proteção implementados são eficazes. O instrumento estará em conformidade se não forem constatadas vulnerabilidades.

**8.3.5.2** Os meios de proteção devem utilizar recursos lógicos e/ou físicos para mitigar as vulnerabilidades, isto inclui selagem através de lacres mecânicos.

### 8.3.6 Análise textual

**8.3.6.1** Avaliar a hierarquização do conteúdo da documentação apresentada pelo requerente buscando identificar se as soluções tecnológicas apresentadas atendem os requisitos do RTM.

## 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, da 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

**8.4.2.1** Revisão do código-fonte na busca de uma determinada informação para averiguar o atendimento a um requisito do RTM e/ou verificar a veracidade de uma informação contida na documentação.

### 8.4.3 Análise do fluxo de dados

**8.4.3.1** Verificar se os intervalos de valores das variáveis do programa estão respeitando seus limites mínimos e máximos. Caso existam intervalos que não respeitem os limites, inspecionar o comportamento e se violam o funcionamento do instrumento. O instrumento estará em conformidade com os requisitos do RTM, se os intervalos que não respeitam os limites não produzem qualquer efeito sobre as funções do instrumento.

### 8.4.4 Análise do fluxo de controle

**8.4.4.1** Verificar se o fluxo lógico do *software* está de acordo com o aspecto dinâmico (fluxo de execução) especificado na descrição de cada bloco do instrumento. Caso existam diferenças entre o fluxo lógico e o de execução, inspecionar o comportamento do instrumento. O instrumento estará em conformidade com os requisitos do RTM se as diferenças, caso existam, não produzem 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 instrumento estará em conformidade se os comandos estiverem alinhados aos requisitos do RTM.

**8.4.5.2** Inspecionar o código-fonte em busca de comandos não descritos. O instrumento 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

**8.4.6.1** Identificar as variáveis legalmente relevantes no código-fonte, bem como seus intervalos de valores. O instrumento de medição de velocidade estará em conformidade se o intervalo de valores para cada variável relevante for válido.

**8.4.6.2** Realizar o rastreamento dessas variáveis ao longo do código-fonte. O instrumento estará em conformidade se os procedimentos que manipulam as variáveis estiverem permitidos e se a implementação desses procedimentos estiver refletida no aspecto dinâmico (fluxo de execução) estabelecido na descrição de cada bloco do instrumento.

**8.4.6.3** As variáveis legalmente relevantes somente podem ser manipuladas através de acesso ao módulo legalmente relevante por usuários com permissão específica, e deverá deixar registros lógicos (registro de auditoria / log) e / ou físicos (rompimento de selagem).

### 8.4.7 Análise de vulnerabilidades

**8.4.7.1** Realizar a análise de possíveis condições provenientes de erros de implementação das interfaces. O instrumento estará em conformidade se não forem constatados erros de implementação das interfaces, diminuindo as possibilidades de intrusão por parte de um atacante.

**8.4.7.2** Considerar as diferentes formas de implementação com suas peculiaridades, de acordo com a solução tecnológica aplicada em cada instrumento, com o objetivo de atender os requisitos do RTM. Arquiteturas similares podem possuir soluções tecnológicas diferentes para se atingir o mesmo objetivo.

**8.4.7.3** Caso a rede de comunicação entre dispositivos metrologicamente relevantes dispor acesso a outros dispositivos (ou usuários) sem rompimento de lacres, esta rede deve ser verificada pelo módulo legalmente relevante utilizando ferramentas de varredura a fim de verificar/identificar modificações de arquitetura ou portas não devidamente protegidas ou serviços ocultos ou não declarados.

**8.4.7.4** Nos casos em que existam duas ou mais interfaces de rede em uma mesma plataforma (placa CPU) deve-se constatar o isolamento entre estas redes, caso existam funcionalidades e/ou recursos legalmente relevantes e não relevantes separados por esta premissa.

**8.4.7.5** O uso de assinatura digital de informações do instrumento deve ser sempre acompanhado de uma fase de verificação da informação assinada, quando esta atingir seu ponto de publicação ou entrega.

**8.4.7.6** Quando for possível o requerente renunciar a soluções de segurança da informação para proteção de informações dentro da cadeia legalmente relevante (autenticidade/integridade) e substituir pela utilização de lacres físicos, estes deverão ser da categoria primário.

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

### 8.5 Ensaio funcional de *software* (EF)

**8.5.1** Consiste na análise do comportamento do *software* legalmente relevante do instrumento em situações de operação real ou emuladas em laboratório.

**8.5.2** O ensaio funcional de *software* deve ser aplicado, preferencialmente após conclusão da análise da documentação e análise do código-fonte para assegurar, ratificar ou respaldar o entendimento do técnico às declarações contidas na documentação avaliada e constatar o completo 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 e dispositivos para simulação de medição fornecidos pelo requerente.

**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 deve ser avaliada. Para interfaces gráficas de usuário, todos os menus e demais elementos gráficos devem ser ativados e avaliados.

**8.5.6** Apesar do RTM não ser explicito, o Inmetro entende que todo instrumento de medição deve armazenar registros de auditoria (logs). Durante o ensaio funcional deve-se verificar se ao menos os

registros de auditoria referente à carga de *software* (tentativas e cargas bem-sucedidas), mudança de parâmetro legalmente relevante e interrupção de funcionamento (incluindo manutenções) são feitos em memória não volátil em ordem cronológica.

## 9 REQUISITOS GERAIS

### 9.1 Características básicas do medidor de velocidade de veículos automotores

**9.1.1** Avaliar se o instrumento atende os requisitos do item 3.1.1 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.1.2** Avaliar a completude da documentação e necessidade de atendimento a requisitos específicos contidos no respectivo RTM.

#### 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** Inspecionar o código-fonte e verificar sua completude.

**9.1.4.2** Na documentação fornecida pelo requerente deve constar uma descrição técnica de como gerar os executáveis embarcados no instrumento.

**9.1.4.2.1** Siga essa documentação e confira arquivos de compilação automática, como *makefile*.

**9.1.4.3** Itens de terceiros (*hardware* ou *software* de prateleira) com código-fonte fechado não necessitam avaliação de código-fonte.

#### 9.1.5 Ensaios funcionais para características básicas do instrumento

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

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

**9.1.5.2** Verificar se a interface de entrada de usuário se comporta do modo especificado na documentação fornecida pelo requerente, em especial os itens relacionados a autenticação, acesso e níveis de permissão, mas não se restringindo a esses.

### 9.2 Identificação/integridade do *software*

**9.2.1** Avaliar se o instrumento atende os requisitos do item 3.1.2 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março 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** Itens de terceiros (*hardware* ou *software* de prateleira) devem ter suas versões de *software* declaradas e verificadas pelo *software* legalmente relevante do instrumento em análise, sem, contudo, necessitar de uma verificação de integridade.

### 9.3 Avaliação da documentação para identificação/integridade de *software*

**9.3.1.1** Avaliar textualmente se a documentação descreve como o identificador é construído, como é estruturado e como pode ser visualizado no instrumento.

**9.3.1.2** Avaliar textualmente se o requerente apresentou em seu memorial descritivo de *software* todas as versões de *software*/*firmware* utilizadas por funções legalmente relevantes, como câmeras, detectores de velocidade, *firewalls*, componentes de rede, etc.

**9.3.1.3** Avaliar textualmente se a documentação mostra de forma inequívoca a versão atual de cada *software* embarcado.

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

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

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

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

**9.3.1.4** No caso de um *software* oriundo de uma modificação de modelo, avaliar textualmente se a documentação apresenta corretamente o identificador de *software* alterado, segundo a regra de construção do identificador apresentada pelo requerente.

**9.3.1.5** 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.

**9.3.1.6** 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.

**9.3.1.7** Caso o instrumento implemente uma metodologia de desafio resposta utilizando o protocolo de comunicação descrito na NIT-Sinst-020, não há obrigatoriedade 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* desenvolvido pelo Inmetro.

**9.3.1.8** Avalie textualmente se a documentação descreve a fonte dos números aleatórios utilizados. Se forem empregados números pseudoaleatórios, o algoritmo utilizado deve ser apresentado junto à documentação e atender os requisitos da norma SP 800-22.

Nota - A documentação pode referenciar uma fonte na literatura que comprove o sucesso do algoritmo nos testes estatísticos propostos na SP 800-22, ou apresentar os resultados dos testes realizados.

**9.3.1.9** As áreas de memórias para armazenamento de dados devem estar indisponíveis durante a verificação de integridade. Avalie textualmente se a documentação descreve a não disponibilidade das áreas de armazenamento durante a verificação de integridade. Por exemplo, caso o instrumento possua memória de massa, essa deve estar indisponível (por exemplo, removida fisicamente) durante o procedimento de verificação de integridade por desafio-resposta.

**9.3.1.9.1** 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.

**9.3.1.10** Na exceção de 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.

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

Nota – Não é raro que a alternativa à 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.

**9.3.2** No caso de uso de sistema operacional, avaliar textualmente se a metodologia proposta pelo requerente verifica a integridade de todos os arquivos que compõe o sistema operacional.

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

**9.3.2.2** Versões de *software*/*firmware* de itens de terceiros diferentes das previamente documentadas devem ser rejeitadas pelo instrumento e uma informação de falha deve ser apresentada no *software* principal e inserida na trilha de auditoria.

**9.3.2.3** Itens que possuam funcionalidade de assinatura digital de suas informações devem ter seus certificados previamente cadastrados mediante rompimento de lacres primários e/ou mediante uso de autenticação de usuário com registro em trilha de auditoria para efetivarem seu funcionamento.

**9.3.2.4** Os itens com funcionalidades de assinatura digital não podem ser utilizado sem ter seu certificado cadastrado.

Nota – Algumas soluções tecnológicas, com a de computação confiável (*trusted computing*), podem garantir a integridade do *firmware* da placa mãe, *bootloader* e do sistema operacional.

### 9.3.3 Análise de código-fonte para identificação/integridade de *software*

**9.3.3.1** Rastrear as variáveis relevantes para o identificador de *software*.

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

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

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

### 9.3.4 Ensaios funcionais para identificação/integridade de *software*

**9.3.4.1** Deve ser seguido o procedimento descrito na documentação encaminhada pelo requerente e verificar se é possível acessar o identificador de versão de *software*.

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

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

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

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

**9.3.4.5** No caso de instrumento em que o protocolo de comunicação da NIT-Sinst-020 não tenha condições técnicas de ser implementado, o requerente deve fornecer uma ferramenta de verificação e integridade.

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.

**9.3.4.6** O requerente deve fornecer, além do *firmware* íntegro, um *firmware* não íntegro a ser utilizado na verificação de integridade.

**9.3.4.7** Verificar se a ferramenta de verificação de integridade consegue distinguir adequadamente um *firmware* íntegro de um *firmware* não íntegro.

**9.3.4.8** 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.

**9.3.4.9** No caso de uso de sistema operacional, verifique se a solução proposta pelo requerente garante a integridade do *bootloader* do sistema operacional, fornecendo o ponto de partida de uma cadeia de confiança. Verificar se os demais pontos da cadeia de confiança de inicialização do sistema operacional são garantidos pelos precedentes e garantem os subsequentes, sem possibilidade de intrusão indevida no processo de inicialização.

### 9.4 Exatidão dos algoritmos e funções de medição

**9.4.1** Avaliar se o instrumento atende os requisitos do item 3.1.3 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.4.2** Os algoritmos e funções de medição devem ser adequados e funcionalmente corretos para o instrumento (precisão dos algoritmos, arredondamentos).

**9.4.3** Deve ser possível analisar algoritmos e funções, tanto por ensaios metrológicos como por exames de *software.*

**9.4.4** O requerente deve fornecer um dispositivo simulador de velocidades que permita avaliar o desempenho do instrumento em várias velocidades simuladas de um veículo automotor.

### 9.4.5 Avaliação da documentação para exatidão dos algoritmos e funções de medição

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

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

**9.4.6.1** Analisar se o fluxo de controle das rotinas implementadas no código-fonte corresponde às rotinas de medição documentadas.

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

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

**9.4.6.4** Rastrear as variáveis utilizadas para armazenar dados de medição.

### 9.4.7 Ensaios funcionais para exatidão dos algoritmos e funções de medição

**9.4.7.1** Checar se os erros de medição da medição de velocidade, nos pontos definidos pelo simulador fornecido pelo fabricante, atendem aqueles do RTM aprovado pela Portaria Inmetro nº 158 de 31 de março de 2022.

Nota – Instrumentos estáticos, portáteis ou móveis podem requerer ensaio de campo para verificação da exatidão dos algoritmos de medição. (e.g., efeito cosseno).

### 9.5 Influência da interface de entrada de dados

**9.5.1** Avaliar se o instrumento atende os requisitos do item 3.1.4 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março 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 do *software* legalmente relevante, de parâmetros legalmente relevantes ou de dados de medição. Não pode haver comandos não declarados.

### 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.

Nota 1 – Caso os comandos permitam alteração no *software* legalmente relevante ou nos parâmetros legalmente relevantes, confirme a existência de autenticação ou rompimento de selagem principal, conforme itens 9.8 e 10.3 da presente norma, respectivamente.

Nota 2 – Caso os comandos trafeguem por rede de comunicação interna, atentar para os requisitos do item 10.2 da presente norma.

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

**9.5.3.3** 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.

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

**9.5.3.5** 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).

Nota 1 – É comum nas arquiteturas de medidores de velocidade a presença de um computador não legalmente relevante utilizado para a lógica de negócio não metrológica (e.g., detectar avanço de sinal). Esse computador legalmente não relevante deve ser analisado como um possível vetor de ataque.

Nota 2 – 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 instrumento através de comandos de interface.

Nota – Realizar a intrusão pode ser uma tarefa que consome tempo. O objetivo desse ensaio é avaliar o exemplar sob ensaio e a possibilidade de realizar a intrusão.

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

### 9.6 Proteção contra mudanças acidentais/não intencionais

**9.6.1** Avaliar se o instrumento atende os requisitos do item 3.1.5 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

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

**9.6.2.1** Pode ser inviável evidenciar esse requisito para alguns *softwares* embarcados em *hardware* de prateleira (e.g., *firmware* de sensor *doppler).* Nesse outro *firmware* do instrumento deve-se manter controle dos parâmetros legalmente relevantes e detectar eventuais falhas reportadas pelos *firmwares* de itens de prateleira.

### 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 acidentais/não intencionais.

**9.6.3.1.1** 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 indique falha, o instrumento deve disparar um comportamento de falha pré-definido na documentação.

**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 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/não intencionais.

### 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, 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 carregá-los no instrumento, para serem utilizados nos ensaios funcionais descritos nos itens 9.6.5.2 até 9.6.5.5 da presente 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 nessas ações.

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

**9.6.5.4** Verificar se o instrumento reconhece e como reage às 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 da presente norma.

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

### 9.7 Proteção contra mudanças intencionais não autorizadas

**9.7.1** Avaliar se o instrumento atende os requisitos do item 3.1.6 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

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

Nota 1 – Caso seja permitido carga de *software*, atente-se para o item 10.3 desta norma.

Nota 2 – Caso seja permitido alteração de parâmetros legalmente relevantes por interface de comunicação, atente-se para o item 9.8 desta norma.

Nota 3 – As medidas de proteção adotadas podem ser mecânicas, eletrônicas, lógicas ou um conjunto dessas.

**9.7.2.1** A arquitetura do requerente deve ser tal que o *hardware* de prateleira, assim como seu *firmware*, fique protegido de mudanças intencionais.

### 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*, dos parâmetros legalmente relevantes e da memória física do instrumento.

**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 eles não estejam acessíveis na versão de produção do 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** Em arquiteturas que possuam registradores para configuração de dados legalmente relevantes, esses podem permitir alterações intencionais da memória do instrumento. Inspecionar se os registradores estão corretamente configurados a fim de impedir alterações intencionais não autorizadas.

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.

**9.7.4.7** Verificar como é realizada a proteção física da memória do instrumento, de modo que não possa ser substituída indevidamente sem o rompimento de selagem primária.

### 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 instrumento ou proteção física que impeça a alteração de características legalmente relevantes do *firmware* por pessoas não autorizadas.

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

**9.7.5.3** Verificar se o plano de selagem apresentado na documentação (Memorial descritivo) 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 – Enviar os comandos de teste quando o instrumento estiver em operação normal e verificar se os mesmos se encontram habilitados. Avaliar se o instrumento em modo de operação normal realiza funções específicas do modo de teste.

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. Outra forma de averiguar uma dissonância entre o código-fonte e o binário é comparar as *strings* presentes 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 de configuração

**9.8.1** Avaliar se o instrumento atende os requisitos do item 3.1.7 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.8.2** Os parâmetros que fixam as características legalmente relevantes do instrumento de medição de velocidade só podem ser modificados se autorizados.

### 9.8.3 Avaliação da documentação para proteção dos parâmetros de configuração

**9.8.3.1** Avaliar textualmente a documentação a respeito da proteção dos parâmetros legalmente relevantes.

**9.8.3.2** Avaliar textualmente a documentação a respeito da visualização dos parâmetros legalmente relevantes.

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

**9.8.3.4** 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*).

**9.8.3.5** 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 primária ou autenticação.

### 9.8.4 Análise de código-fonte para proteção dos parâmetros

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

### 9.8.5 Ensaios funcionais para proteção dos parâmetros

**9.8.5.1** Caso exista um modo de configuração que permita 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 permita alterar parâmetros legalmente relevantes, alterne o instrumento para o modo de configuração e realize uma alteração dos parâmetros legalmente relevantes. Retorne ao modo de operação. Verificar se ao retornar ao modo de operação o instrumento está utilizando os novos parâmetros.

### 9.9 Detecção de falha

**9.9.1** Avaliar se o instrumento atende os requisitos do item 3.1.8 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março 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 já previstas no projeto e se ele toma as decisões também já previstas diante dessas falhas.

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

**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, coloque 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 que interfira no tratamento das informações legalmente relevantes, verificar se o processo de medição foi interrompido.

Nota – Dependendo da arquitetura do instrumento, é possível que alguns estados de falha não possam ser reproduzidos na bancada.

## 9.10 Validação de *software*

**9.10.1** Avaliar se o instrumento atende os requisitos do item 3.1.9 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

Nota – Os casos de teste documentados pelo requerente são úteis para montagem experimental dos ensaios funcionais propostos nesta 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 todos 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 instrumento antes do início do teste. |
| 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 | São as ações que o usuário deve fazer para que o instrumento 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* ou foto da tela do instrumento contendo o resultado, registro fotográfico ou gravação de vídeo, arquivo de log do instrumento, bloco de dados trafegado como resposta, etc. |
| Pós-condições | Para cada condição de execução, descreve o estado ao qual o instrumento deverá retornar para permitir a execução de testes subsequentes. Relatar somente em casos excepcionais. |

Fonte: Sinst

### 9.10.3 Composição do resultado da medição da velocidade

**9.10.3.1** Avaliar se o instrumento atende os requisitos do item 3.1.10 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

### 9.10.3.2 Avaliação da documentação para composição do resultado da medição da velocidade

**9.10.3.2.1** Avaliar textualmente se a documentação descreve se a composição do resultado da medição de velocidade contém os dados descritos no item 3.1.10 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.10.3.2.1.1** Os dados de medição devem conter o registro fotográfico. A definição de registro fotográfico (que inclui mais informações do que meramente a foto) está nos itens 3.11 e 3.12 do Anexo A do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.10.3.2.2** Estas informações podem estar contidas em um único arquivo jpeg, por exemplo, ou contida em um pacote compactado do tipo .zip, .tar, etc. Estas informações devem ser submetidas posteriormente a uma ferramenta (*software*) que ateste a integridade, autenticidade e não repúdio das informações contidas no arquivo ou no conjunto de arquivos.

**9.10.3.2.3** Avaliar textualmente se o instante de tempo da medição de velocidade é obtido por meio de um relógio sincronizado com uma referência de tempo UTC.

**9.10.3.2.4** Avaliar textualmente se o relógio apresenta uma deriva máxima de 1 minuto em um período de 30 dias em relação à referência de tempo UTC.

**9.10.3.2.5** Avaliar textualmente se funções e variáveis envolvidas no cálculo e composição do resultado da medição estão identificadas e suficientemente descridas.

**9.10.3.3 Análise de código-fonte para composição do resultado da medição da velocidade**

**9.10.3.3.1** Avaliar textualmente se a documentação descreve se a composição do resultado da medição de velocidade contém os dados descritos no item 3.1.10 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.10.3.3.2** Analisar o código-fonte para verificar se o instante de tempo da medição de velocidade é obtido por meio de um relógio sincronizado com uma referência de tempo UTC.

**9.10.3.3.3** Analisar o código-fonte para verificar se o relógio apresenta uma variação máxima de 1 minuto em relação à referência de tempo UTC, com deriva temporal correspondente a, no máximo, 1 minuto em um período de 30 dias.

**9.10.3.3.4** Analisar o código-fonte para verificar se o relógio apresenta uma rotina de atualizações do tipo microcorreções associadas a uma base de tempo do tipo NTP *server* ou a um dispositivo GPS. O requerente deve declarar qual o desvio de tempo que será considerado como uma microcorreção.

**9.10.3.3.5** Valores de correção temporal superiores aos declarados como microcorreção devem estar protegidos por autenticação e considerados como legalmente relevantes.

**9.10.3.3.6** Analisar o código-fonte para verificar se as variáveis que influenciam a composição do resultado da medição são acessadas apenas pelas funções, comandos e trechos de código documentados.

**9.10.3.4 Ensaios funcionais para composição do resultado da medição da velocidade**

**9.10.3.4.1** Obter um pacote de dados de medição e verificar se ele compreende a composição do resultado da medição de velocidade e contém os dados descritos no item 3.1.10 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.10.3.4.2** Avaliar se o meio declarado pelo requerente para efetuar sincronismo de seu relógio interno está funcionando conforme esta declaração.

**9.10.3.4.3** Avaliar se o instante de tempo da medição de velocidade é obtido por meio de um relógio sincronizado com uma referência de tempo UTC.

**9.10.3.4.4** Avaliar se o relógio apresenta uma variação máxima de 1 minuto em relação à referência de tempo UTC.

**9.10.3.4.5** Avaliar se o relógio deve apresentar uma deriva temporal correspondente a, no máximo, 1 minuto em um período de 30 dias.

**9.10.3.4.6** Avaliar se o pacote de dados de medição está protegido contra modificação através da manipulação de arquivos, variáveis do sistema, ou qualquer outro recurso, que provoque alteração no resultado da medição de velocidade. Esta proteção consiste em impedir ou identificar e evidenciar a modificação.

### 9.10.4 Autenticidade e integridade do resultado de medição

**9.10.4.1** Avaliar se o instrumento atende os requisitos do item 3.1.11 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

### 9.10.4.2 Avaliação da documentação para autenticidade e integridade do resultado de medição

**9.10.4.2.1** Avaliar textualmente se o resultado da medição é protegido contra mudanças intencionais e se permite, *a posteriori*, remontar sua origem por meio de mecanismo de assinatura digital reconhecidamente seguro constante do documento FIPS PUB 186-4 - *Digital Signature Standard (DSS)*.

**9.10.4.2.2** Avaliar textualmente a correta atribuição de um par de chaves pública/privada, que pode ser garantido via uma ICP (Infraestrutura de Chave Pública) e neste caso deve ser descrita toda a cadeia de certificação empregada.

**9.10.4.2.3** Avaliar textualmente, caso não haja a utilização de autoridades certificadoras, se são utilizados outros meios e a eficácia destes meios em garantir a correspondência dos pares de chaves empregados.

**9.10.4.2.4** Avaliar textualmente se o fabricante utiliza algoritmos de geração de chaves adequados, as medidas de proteção e sigilo da chave privada até sua introdução no instrumento, e o mecanismo que disponibiliza a chave pública do instrumento.

**9.10.4.2.5** Avaliar textualmente se a assinatura digital do resultado da medição contempla todos os dados referenciados no item 3.1.10 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.10.4.2.6** Avaliar textualmente se os dados do resultado da medição, assim como suas assinaturas digitais, são arquivados e mantidos em conjunto. Avaliar textualmente se os dados do resultado da medição legalmente relevante são disponibilizados para conferência da integridade, quando solicitado.

**9.10.4.3 Análise de código-fonte para autenticidade e integridade do resultado de medição**

**9.10.4.3.1** Analisar, no código-fonte, se o resultado da medição é protegido contra mudanças intencionais por meio de mecanismo de assinatura digital constante do documento FIPS PUB 186-4 - *Digital Signature Standard (DSS).*

**9.10.4.3.2** Analisar, no código-fonte, se a correta atribuição de um par de chaves pública/privada pode ser garantida via uma ICP e se é descrita toda a cadeia de certificação empregada.

**9.10.4.3.3** Caso não haja a utilização de autoridades certificadoras, analisar no código-fonte quais os meios previstos e a eficácia deste meio em garantir a correspondência dos pares de chaves empregados.

**9.10.4.3.4** Analisar no código-fonte se a assinatura digital do resultado da medição contempla todos os dados explicitados no item 3.1.10 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.10.4.4 Ensaios funcionais para autenticidade e integridade do resultado de medição**

**9.10.4.4.1** Avaliar se o pacote de medição contempla todos os dados explicitados nos itens 3.1.10 e 3.1.11 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 20224 e se é assinado digitalmente por um algoritmo de assinatura digital reconhecidamente seguro constante do documento FIPS PUB 186-4 - *Digital Signature Standard* (DSS).

**9.10.4.4.2** Verificar a assinatura digital de um pacote de medição assinado pelo instrumento utilizando a cadeia de certificação empregada.

**9.10.5 Vínculo entre a medição e o registro fotográfico**

**9.10.5.1** Avaliar se o instrumento atende os requisitos do item 3.1.12 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.10.5.2 Avaliação da documentação para vínculo entre a medição e o registro fotográfico**

**9.10.5.2.1** Avaliar textualmente se há algum mecanismo que garanta a correta vinculação entre a medição de velocidade e a obtenção do registro fotográfico, e se esse mecanismo é efetivo.

**9.10.5.2.2** O plano de selagem (primário) deve evidenciar a desconexão do cabo ethernet nos terminais da câmera, do módulo legalmente relevante, e quaisquer outros dispositivos que estejam neste caminho. Sendo dispensado nos casos em que o módulo legalmente relevante verificar que o registro fotográfico foi assinado digitalmente pela câmera.

**9.10.5.2.3** Nos casos em que for utilizada assinatura digital da câmera é necessário que o cadastro de chave pública da câmera no módulo legalmente relevante seja permitido apenas após confirmação de usuário e registro de auditoria.

**9.10.5.2.4** Para utilização de assinatura digital da câmera na autenticação do registro fotográfico, os algoritmos implementados devem ser informados na documentação, e não devem existir notificações de vulnerabilidades sobre os mesmos.

**9.10.5.3 Análise de código-fonte para vínculo entre a medição e o registro fotográfico**

**9.10.5.3.1** Avaliar no código-fonte se o mecanismo que garante a correta vinculação entre a medição de velocidade e a obtenção do registro fotográfico está adequadamente implementado e corresponde à descrição constante da documentação.

**9.10.5.3.2** Avaliar se os algoritmos envolvidos na assinatura digital e sua verificação estão de acordo com a documentação e as exigências do regulamento técnico metrológico de medidores de velocidade.

**9.10.5.4 Ensaios funcionais para vínculo entre a medição e o registro fotográfico**

**9.10.5.4.1** Obter pacote de dados com vinculação entre a medição de velocidade e a obtenção do registro fotográfico e verificar se o mecanismo documentado e implementado é efetivo.

**9.10.5.4.2** Verificar se o sistema é capaz de impedir ou rejeitar alterações no registro fotográfico ou em seu vínculo.

**9.10.5.4.3** Avaliar e relatar se existe alguma forma de substituir a câmera sem deixar evidências.

**9.10.5.4.4** Avaliar se existem outras partes do sistema não documentadas em que um pacote de dados ou um registro fotográfico podem ser introduzidos sem deixar evidências.

**9.10.6 Confidencialidade das chaves**

**9.10.6.1** Avaliar se o instrumento atende os requisitos dos itens 3.1.12.2, 3.1.12.3 e 3.1.12.4 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**9.10.6.2 Avaliação da documentação para confidencialidade das chaves**

**9.10.6.2.1** Avaliar textualmente se as chaves criptográficas privadas utilizadas são tratadas como dados legalmente relevantes e são mantidas em segredo e protegidas contra quaisquer possibilidades de comprometimento.

**9.10.6.2.2** Avaliar textualmente se as chaves privadas são protegidas por meio de selagem e, caso contrário, se há alguma proteção.

**9.10.6.3 Avaliação da documentação para confidencialidade das chaves**

**9.10.6.3.1** Avaliar no código-fonte se as chaves criptográficas utilizadas são protegidas como dados legalmente relevantes, mantidas em segredo e protegidas contra quaisquer possibilidades de comprometimento.

**9.10.6.3.2** Avaliar no código-fonte se as chaves privadas possuem alguma proteção por *software*, caso não tenha proteção por meio de selagem.

**9.10.6.4 Ensaios funcionais para confidencialidade das chaves**

**9.10.6.4.1** Verificar se há possibilidade de extração ou comprometimento das chaves privadas através do acesso ao instrumento.

## 10 REQUISITOS ESPECÍFICOS

### 10.1 Separação das partes legalmente relevantes

**10.1.1** Avaliar, se pertinente, se o instrumento atende os requisitos do item 3.2.1 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**10.1.2** Os requisitos do item 3.2.1 citado devem ser cumpridos com intuito de comprovar a separação inequívoca do *software* em *software* legalmente relevante e *software* não legalmente relevante. Se não existir separação de *software*, todo o *software* deve ser considerado como legalmente relevante.

Nota – É comum nas arquiteturas de medidores de velocidade a presença de *software* não legalmente relevante utilizado para a lógica de negócio não metrológica (e.g., detectar avanço de sinal).

### 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 instrumento identificando as partes legalmente relevantes e não legalmente relevantes de forma clara.

**10.1.3.1.2** Descrição de todas as funções de programa e estruturas de dados legalmente relevantes. Não deverá existir qualquer 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* não legalmente 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 possíveis de serem usados 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* não legalmente relevante estão 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* não legalmente 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 instrumento pode ter *software* não legalmente 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 instrumento (entre o *software* legalmente relevante e o *software* não legalmente 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** Certificar-se de que nenhum aspecto legalmente relevante foi implementado na parte não legalmente relevante.

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

**10.1.3.5** Analisar o esquemático de *software* e *hardware* e comparar com o diagrama de blocos que compõe o instrumento e a interface de *software* protetora. Ambos devem evidenciar a separação da parte legalmente relevante da parte não legalmente relevante. O *software* não legalmente relevante não pode alterar a memória do *software* legalmente relevante.

**10.1.3.6** Avaliar a existência de eventuais vulnerabilidades na interface protetora de *software*.

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

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

**10.1.4.2** Analisar a completude de comandos da interface protetora de *software*, identificando se todos os comandos declarados na documentação estão implementados no código-fonte e se há comandos não declarados na documentação implementados no código-fonte.

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

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

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

Nota – Nas arquiteturas com separação de *software* não é raro que o *software* não legalmente 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.

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

**10.1.4.7** 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 adequada 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 àquele documentado.

**10.1.5.2.1** Investigar a existência de interações inadmissíveis entre as partes legalmente relevantes e não legalmente relevantes.

**10.1.5.2.1.1** Enviar para o instrumento comandos identificados no código-fonte como *debug*, teste, ou que funcionam em modos especiais de operação, o instrumento não deve executá-los. Comandos não documentados que forem identificados na avaliação do código fonte devem ser enviados para garantir que o software foi de fato atualizado para inibi-los.

**10.1.5.3** Caso a apresentação de informações 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 se a carga de *software* não legalmente 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 atende os requisitos do item 3.2.2 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**10.2.2** O conjunto de requisitos técnicos descritos a seguir se aplica apenas quando o instrumento utiliza, internamente à cadeia de medição legalmente relevante, 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 dos dados de medição são transmitidas.

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

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

Nota – Algumas vezes é utilizada 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 3.2.3 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022, verificar por análise textual da documentação se os algoritmos apresentados estão de acordo com o documento SP 800-57 parte 1.

Nota – Caso a arquitetura use ferramentas de criptografia para garantir integridade e autenticidade de informação trafegada na rede interna deve-se estar atento para a trilha de auditoria no caso de acesso físico aos componentes da rede.

**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 sua extração do meio físico ou lógico.

**10.2.3.5.2** Caso o instrumento faça uso de chaves criptográficas, analisar textualmente as proteções contra mudanças não intencionais delas.

**10.2.3.5.3** Caso o instrumento faça uso de chaves criptográficas, analisar textualmente as proteções contra mudanças intencionais delas.

**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 atraso 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** As redes de comunicação referentes aos *softwares* legalmente relevante e não legalmente relevante devem ser claramente identificadas, a documentação deve especificar os elementos e medidas protetivas da parte referente ao software legalmente relevante.

**10.2.3.10** Analisar textualmente se as medidas protetivas são eficazes para detectar e reagir à inclusão de elementos na rede do *software* legalmente relevante sem a devida autorização e registro de auditoria.

Nota – Parar de realizar medições é uma reação considerada adequada quando o instrumento consegue detectar a inclusão de um elemento na rede, mas não é capaz de identificar, e controlar sua influência.

**10.2.3.11** Nos casos em que o gerenciador da rede (roteador / *switch*) é utilizado para as redes legalmente relevante e não legalmente relevante, é preciso que a documentação demonstre os recursos utilizados para que o software legalmente relevante seja capaz de identificar alterações na configuração do gerenciador (roteador / *switch*), e disparar medida protetiva para impedir interferência na rede metrologicamente relevante. O gerenciador (roteador / *switch*) deve ser considerado legalmente relevante recebendo selagem primária.

**10.2.3.12** Analisar eventuais vulnerabilidades no uso das primitivas criptográficas.

Nota – Uma boa prática no uso de primitivas criptográficas é utilizar código que já foi amplamente testado. Verifique 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 intencionais e 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, a 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 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 instrumento.

### 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 instrumento é realizada de acordo com o especificado na documentação fornecida.

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

**10.2.5.2** Quando houver o compartilhamento de um dispositivo de rede para gerenciamento (roteador / switch) das redes metrologicamente relevante e metrologicamente não relevante, é necessário que sua configuração seja monitorada pelo software metrologicamente relevante, de forma que qualquer modificação na rede metrologicamente relevante seja detectada, iniciando a medida protetiva documentada.

**10.2.5.3** Se aplicável, verificar se a troca de posição/ordem da conexão dos componentes do instrumento é detectada. Verificar se esta detecção provoca a reação descrita na documentação.

**10.2.5.4** Verificar se a reação do instrumento a atrasos, interrupções e indisponibilidade de serviços é aquela descrita na documentação fornecida pelo requerente.

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

**10.2.5.6** 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 atende os requisitos do item 3.2.3 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março 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 norma NIT-Sinst-003 item 11.

**10.3.3.1.1** Para cada algoritmo listado na norma FIPS PUB 186-4 - *Digital Signature Standard (DSS)*, 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* legalmente relevante 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 norma 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*.

**10.3.3.3** Analisar textualmente se a carga e a subsequente instalação de *software* são automáticas e garantem o não comprometimento do ambiente de proteção do *software* no final do processo.

**10.3.3.4** Analisar textualmente se o instrumento tem um *software* legalmente relevante permanentemente residente e invariável, com todas as funções necessárias para verificar os requisitos de carga.

**10.3.3.5** Analisar textualmente se o instrumento é capaz de detectar uma falha de carga ou instalação, gerando uma sinalização do ocorrido.

**10.3.3.6** Analisar textualmente, caso a carga ou a instalação fracassar, ou se for interrompida, se o estado inicial do instrumento não é afetado.

**10.3.3.7** Analisar textualmente, caso a carga ou instalação fracassar e não seja possível retornar ao estado inicial, se o instrumento exibe uma mensagem de erro permanente, e seu funcionamento metrológico é impedido, até que o erro seja corrigido.

**10.3.3.8** Analisar textualmente se, no caso de uma instalação bem sucedida, todas as formas de proteção são restauradas para o seu estado original, a menos que o *software* carregado tenha a devida autorização para alterá-las.

**10.3.3.9** Analisar textualmente se, durante a carga e a instalação de novo *software,* as funções de medição do instrumento são impedidas, caso não possam ser completamente garantidas.

**10.3.3.10** Analisar textualmente se são empregados meios para garantir a autenticidade do *software* carregado (tal como a assinatura digital do *software* legalmente relevante) e para indicar que este *software* foi previamente avaliado e aprovado pelo Inmetro (tal como a realização da assinatura digital pelo Inmetro).

**10.3.3.11** Analisar textualmente se, antes da utilização do *software* carregado, o instrumento verifica automaticamente se o *software* é autêntico e se é aprovado pelo Inmetro.

**10.3.3.12** Analisar textualmente se os meios pelos quais o *software* identifica a sua autorização prévia são protegidos.

**10.3.3.13** Analisar textualmente se são empregados meios para garantir que o *software* tenha sua integridade verificada e somente possa ser usado se esta for constatada.

**10.3.3.14** Analisar textualmente se a carga de *software* é armazenada em trilha de auditoria contendo: estado de sucesso ou insucesso da carga, carimbo de tempo da carga, versão do *software* antes da carga, versão do *software* após a carga, identificador do operador que realizou a carga.

**10.3.3.15** Analisar textualmente a forma de proteção da trilha de auditoria, que não pode ser apagada ou alterada indevidamente e deve ser protegida como dados legalmente relevantes.

**10.3.3.16** Analisar textualmente se, em caso de separação de *software*, a carga da parte legalmente relevante é protegida de acordo com os itens 10.3.3.1 a 10.3.3.15 da presente norma e se a carga da parte não legalmente relevante do *software* é realizada sem necessidade de controle por parte do Inmetro.

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

**10.3.4.1** Analisar o código-fonte, o fluxo de controle e as rotinas responsáveis por verificar a assinatura digital do *software* a ser carregado.

**10.3.4.2** Analisar o código-fonte, o fluxo de controle e as rotinas responsáveis pelo controle de permissão para realização da carga de *software* legalmente relevante.

**10.3.4.3** Analisar o código-fonte, o fluxo de controle e as rotinas que criam a trilha de auditoria de carga de *software*.

**10.3.4.4** Analisar o código-fonte, o fluxo de controle e as rotinas que gerenciam a carga de *software*, e se detectam seu sucesso ou insucesso e restabelecem os níveis de proteção após a carga.

**10.3.4.5** Analisar o código-fonte, o fluxo de controle e as rotinas que impedem a medição durante a carga de *software* e no caso de seu insucesso.

**10.3.4.6** Analisar o código-fonte, o fluxo de controle e as rotinas que verificam integridade e autenticidade do *software* carregado.

**10.3.4.7** Inspecionar se as medidas descritas na documentação para carga de *software* estão implementadas no código-fonte.

### 10.3.5 Ensaios funcionais para carga de *software* legalmente relevante

**10.3.5.1** Verificar se a carga autorizada (com assinatura digital correta) de *software* legalmente relevante é possível e se é realizada de acordo com a documentação fornecida pelo requerente.

**10.3.5.2** Verificar se a carga não autorizada (com assinatura digital incorreta) de *software* legalmente relevante 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 carga de *software* legalmente relevante não íntegro é detectada e recusada pelo instrumento.

**10.3.5.4** Verificar se a reação à falha quando a carga de *software* legalmente relevante é interrompida antes de sua finalização é sinalizada e se a reação está de acordo com a documentação apresentada pelo requerente.

**10.3.5.5** Verificar se, após uma carga de *software* legalmente relevante com sucesso, as formas de proteção são restauradas para o seu estado original, a menos que o *software* carregado tenha autorização para alterá-las.

**10.3.5.6** Verificar se, durante a carga e a instalação de novo *software* legalmente relevante, as funções de medição do instrumento são impedidas, caso não possam ser completamente garantidas.

**10.3.5.7** 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.

**10.3.5.8** Verificar se os mecanismos de permissão para realização de carga de *software* foram implementados de acordo com a documentação e somente admitem a carga com a permissão explícita do operador do instrumento.

## 11 COMPORTAMENTO DINÂMICO

**11.1** Avaliar se o instrumento atende os requisitos do item 3.2.4 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**11.2** Avaliar se o desempenho do *software* não legalmente relevante não influencia negativamente no comportamento dinâmico do processo de medição executado pelo *software* legalmente relevante.

### 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* não legalmente relevante.

**11.3.2** Avaliar textualmente se, nos casos de testes de *software* realizados pelo fabricante, o instrumento teve comportamento satisfatório, sem que as tarefas do software não legalmente relevante prejudiquem o desempenho das tarefas do *software* legalmente relevante.

**11.3.3** A documentação deve descrever como a hierarquia de interrupções privilegia a execução dos processos legalmente relevantes.

### 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.

Nota – Em instrumento utilizando *Linux* verificar os valores de *priority* e *nice* do *software* legalmente relevante, ações similares devem ser executadas no caso de outros sistemas.

### 11.5 Ensaios funcionais para comportamento dinâmico

**11.5.1** Verificar se o desempenho de processos legalmente relevantes do processador do instrumento é influenciado (diminuído) pela realização de processos não legalmente relevantes.

**11.5.2** Disparar a máxima quantidade de processos não legalmente relevantes, ao mesmo tempo em que se obtêm os dados para compor o resultado da medição da velocidade e processamento das informações legalmente relevantes.

## 12 CAPACIDADE DE PROCESSAMENTO

**12.1** Avaliar se o instrumento atende os requisitos do item 3.2.5 do Anexo B do RTM aprovado pela Portaria Inmetro nº 158, de 31 de março de 2022.

**12.2** Verificar se o dimensionamento dos dispositivos de comunicação compartilhados do instrumento é capaz de atender às necessidades de comunicação entre as partes.

### 12.3 Avaliação da documentação para capacidade de processamento

**12.3.1** Verificar textualmente se a documentação apresenta todos os dispositivos constituintes do instrumento que tenham uso compartilhado (concentradores, redes de comunicação, rádios, *switches*, etc.).

**12.3.2** Verificar textualmente se todos os dispositivos foram dimensionados em função dos instantes de maior fluxo de dados legalmente relevantes.

**12.3.3** 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 instrumento foram dimensionados para os instantes de maior fluxo de dados legalmente relevantes. 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 |
|---|---|---|
| 01 | Abr/2023 | ▪ Atualização dos requisitos da Portaria Inmetro nº 158, de 31/03/2022,<br>▪ Adequação da NIT a NIG-Gabin-040, Rev. 02;<br>▪ Adequação ao MOD-Gabin-040, Rev. 02; e<br>▪ Inclusão de itens: 8.4.6.3 (Procedimentos), 9.1.4.2.1, 9.10.3.2.5, 9.10.3.3.6, 9.10.3.4.6, 9.10.5.2.2, 9.10.5.2.3, 9.10.5.2.4, 9.10.5.4.2, 9.10.5.4.3, 9.10.5.4.4 (Requisitos Gerais), 10.2.3.9, 10.2.3.10, 10.2.3.11, 10.2.3.12, 10.2.5.6 (Requisitos Específicos). |

| Quadro de Aprovação | | |
|---|---|---|
| | **Nome** | **Atribuição** |
| **Elaborado por:** | Juliana Wilm Guedes<br>Alexandre Arosa Saturnino | Auxiliar administrativo<br>Técnico em Metrologia e Qualidade |
| **Verificado por:** | Rogério Possidonio Nunes | Pesquisador Tecnologista em Metrologia e Qualidade |
| **Aprovado por:** | Ícaro dos Santos França | Chefe do Sinst, substituto |