INMETRO	**Instituto Nacional de Metrologia, Qualidade e Tecnologia**

ORIENTAÇÕES PARA REDAÇÃO DE MEMORIAL DESCRITIVO DE SOFTWARE PARA MEDIDORES DE ENERGIA ELÉTRICA

Documento de caráter orientativo

DOQ-DIMEL-009

Revisão 00 - Novembro/2018

SUMÁRIO

1 OBJETIVO

Esse documento tem por finalidade prover informações básicas para a redação do memorial descritivo de software para medidores de energia elétrica. No memorial descritivo de software o requerente explicita como cumpriu os requisitos da Portaria Inmetro 586/2012.

2 CAMPO DE APLICAÇÃO

Se aplica aos requerentes e laboratórios envolvidos no processo de avaliação de software da diretoria de metrologia legal.

3 RESPONSABILIDADE

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

4 DOCUMENTOS DE REFERÊNCIA

Portaria Inmetro nº 371 de 28/09/2007	Aprova os requisitos técnicos e metrológicos como condições mínimas a que devem satisfazer os Sistemas de Medição Centralizada para uso em medição de energia elétrica em unidades consumidoras
Portaria Inmetro nº 586 de 01/11/2012	Aprova o regulamento técnico metrológico de software para medidor eletrônico de energia elétrica e software para sistema distribuído de medição de energia elétrica
Portaria Inmetro nº 587 de 05/11/2012	Aprova o regulamento técnico metrológico para medidores eletrônicos de energia elétrica ativa e/ou reativa, monofásicos e polifásicos, inclusive os recondicionados
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
OIML D 31	*General requirements for software controlled measuring instruments – OIML, 2008*

(Continua)

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*
Ata de reunião Inmetro, 23/6, 2/7 e 3/7, 22/9 de 2015	Segurança de software e interoperabilidade
SP 800-22	A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications
SP 800-57	Recommendation for Key Management, Part 1: General

5 DOCUMENTOS COMPLEMENTARES

NIT-Sinst-003	Organização da documentação para o processo de avaliação de software
NIT-Sinst-019	Análise de software para avaliação de modelo de instrumentos/sistemas de medição de energia elétrica
Plan-Sinst-019	Planilha de Registro de Avaliação de Software para Avaliação de Modelo de Instrumentos/Sistemas de Medição de Energia Elétrica.

6 DEFINIÇÕES

6.1 Siglas

As siglas das UP/UO do Inmetro podem ser acessadas em http://intranet.inmetro.gov.br/tema/qualidade/docs/pdf/siglas-inmetro.pdf.

AM	Avaliação de Modelo
RTM	Regulamento Técnico Metrológico
VIM	Vocabulário Internacional de Metrologia
VIML	Vocabulário Internacional de Metrologia Legal
EFS	Ensaio Funcional de Software
AMD	Análise do Memorial Descritivo
ACF	Análise do Código Fonte
SDMEE	Sistema Distribuído de Medição de Energia Elétrica
TLI	Terminal de Leitura Individual
DVIS	Dispositivo Verificador de Integridade de Software
RTOS	Real Time Operating System
ROM	Read Only Memory

6.2 Termos

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

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

6.2.3 Ataque de força bruta – Método de obter informação não autorizada por tentativa e erro.

6.2.4 Autenticação de integridade – O processo de fornecer garantia de que os dados, ou software, não foram modificados desde que um código de autenticação foi criado para esses dados.

6.2.5 Autenticação da fonte – O processo de fornecer garantia sobre a fonte de informação. Às vezes chamado de autenticação de identidade ou autenticação de origem.

6.2.6 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.

6.2.7 Cadeia legalmente relevante – Todo hardware e software envolvido no processo de medição que consiste na aquisição, processamento e publicação dos dados de medição.

6.2.8 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.

6.2.9 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.

6.2.10 Checksum – Código utilizado para verificar a integridade de dados transmitidos.

6.2.11 Diagrama de Atividade – Fluxograma que demonstra a lógica de funcionamento de um determinado processo. Cada passo do processo, ou atividade, é denotada por uma caixa retangular e cada decisão denotada por um losango.

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

6.2.13 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 6.2.15.

6.2.14 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.

6.2.15 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.

6.2.16 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 considerados construídos 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.
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.

6.2.17 Interface de entrada de usuário – Interface de interação com o instrumento através de um meio físico, como teclado ou touchscreen.

6.2.18 Interface de software protetora – Interface entre o software legalmente relevante e o software legalmente não relevante.

6.2.19 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.

6.2.20 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.

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

6.2.22 Memória de programa – Área de memória aonde está gravado o programa que está sendo executado.

6.2.23 Memorial descritivo – Documento que descreve detalhadamente as implementações tecnológicas voltadas para o atendimento dos requisitos de segurança de hardware e software.

6.2.24 Modo de configuração – Modo que permite alterar parâmetro legalmente relevante.

6.2.25 Não legalmente relevante – Todo software/hardware/dados presentes no instrumento que não são legalmente relevantes.

6.2.26 Parâmetro legalmente relevante – Parâmetro de um instrumento de medição ou de um de seus subconjuntos sujeito ao controle legal.

6.2.27 Parâmetro metrologicamente relevante – Parâmetro legalmente relevante a qual é utilizado no cálculo para se obter o valor da medição. Por exemplo, num medidor tarifa branca data e hora são parâmetros legalmente relevantes, mas não são parâmetros metrologicamente relevantes. Já o Kh é um parâmetro metrologicamente relevante.

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

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

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

6.2.31 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.

6.2.32 Registro de Alteração de Parâmetros Metrológicos Relevantes – Registro de auditoria que armazena os eventos relacionados às alterações de parâmetros metrológicos relevantes no instrumento.

6.2.33 Registro de Cargas de Software Legalmente Relevante – Registro de auditoria que armazena os eventos relacionados às operações de carga de software legalmente relevante no instrumento.

6.2.34 Requerente – É toda 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.

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

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

6.2.37 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.

6.2.38 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.

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

6.2.40 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.

6.2.41 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.

6.2.42 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.

7 MEMORIAL DESCRITIVO DE SOFTWARE

Esse documento orientativo apresenta as diretrizes gerais de como atender os requisitos de software do RTM anexo a Portaria n.º 586, de 01 de novembro de 2012. Entretanto, ainda é uma abordagem generalista não atentando para as peculiaridades de cada instrumento. O requerente, ao submeter a documentação de software, deve ser detalhista e atento para estabelecer claramente um vínculo entre os requisitos do regulamento técnico metrológico anexo a portaria e a arquitetura de seu instrumento.

Nota - A documentação a ser entregue está detalhada na NIT-Sinst-003. Esse documento orientativo versa apenas de um dos itens da documentação, o memorial descritivo de software.

7.1 Elementos pré textuais e textuais

A documentação relacionada aos requisitos de segurança de software e hardware, doravante memorial descritivo de software, deve possuir capa, quadro de controle de revisão de documento, sumário, lista de figuras e lista de tabelas. Deve identificar claramente o modelo em avaliação de software. O documento deve ter páginas e sessões numeradas. O documento deve ser redigido em português. Recomenda-se que o documento seja enviado em pdf não protegido.

Caso o instrumento de medição possua mais de um firmware embarcado, doravante módulo, o memorial descritivo de software deve descrever detalhadamente o atendimento para os requisitos de cada módulo que compõe o sistema de medição. Caso a documentação fique muito extensa e o sistema de medição possua muitos módulos, o memorial de software pode ser dividido em múltiplos documentos, um para cada módulo.

Nota - as devidas proteções para a segurança da informação contida na documentação são descritas na NIT-Sinst-003.

7.2 Requisitos gerais de segurança de software e hardware

Os requisitos gerais compreendem os requisitos de segurança cibernética aplicáveis a todas as arquiteturas de medidores de energia elétrica regulamentados pela Portaria n.º 586, de 01 de novembro de 2012.

7.2.1 Características básicas do sistema/instrumento de medição

O item 7.2.1 elucida o requisito 3.1.1.2 da Portaria n.º 586, de 01 de novembro de 2012.

Devem ser descritos detalhadamente os recursos que suportam a execução do software (Memória, processador/microcontrolador); os aspectos estáticos (arquitetura de software) e dinâmicos (fluxos de execução) do software; funcionalidades específicas do bloco que contribuem para o funcionamento do sistema de medição como um todo (características específicas do mostrador e do medidor, por exemplo), mecanismos físicos de proteção (caixa selada ou não, por exemplo).

Os documentos que descrevem o instrumento estão especificados no item 8 da NIT-Sinst-003. Se for do interesse do requerente, os documentos descritos no item 8.2 e 8.3 da NIT-Sinst-003 podem ser concatenados ao memorial de software. Além dos documentos do item 8 da NIT-Sinst-003, o memorial descritivo de software deve conter as seguintes informações.

7.2.1.1 Identificação da cadeia legalmente relevante

Identificar e descrever a cadeia legalmente relevante, isto é, todos os elementos de software, hardware e dados envolvidos geração da informação relevante para a medição, na captura dos dados de medição, no processamento dos dados de medição e na divulgação do resultado de medição. Apresentar a cadeia legalmente relevante com uso de diagrama de blocos. Observações:

(i) Rotinas de automação que controlem a dinâmica da medição fazem parte da cadeia legalmente relevante.

(ii) Componentes de software que preparem dados legalmente relevantes para armazenamento ou transmissão, ou que realizem a verificação dos dados após leitura ou recepção, pertencem ao software legalmente relevante.

(iii) Rotinas de automação que controlem a dinâmica da medição fazem parte da cadeia legalmente relevante.

7.2.1.2 Plano de selagem

Descrever o plano de selagem e sua relevância para a segurança do equipamento. Os planos de selagem devem ser documentados com o uso de desenho técnico conforme descrito no item 7.6 da NIT-Sinst-003.

7.2.1.3 Detalhamento do processador/microcontrolador

- Descrever as principais características do processador/microcontrolador utilizado.
- Referenciar o manual descritivo (datasheet).
- Detalhar o ambiente de desenvolvimento/programação do processador/micro controlador. Por exemplo, descrever a linguagem de programação utilizada, recursos disponíveis, etc.

7.2.1.4 Memória

Descrever: Tipo, Quantidade, Mapa, etc. A figura 1 mostra um exemplo de mapa de memória.

Figura 1 - Mapa de memória do processador

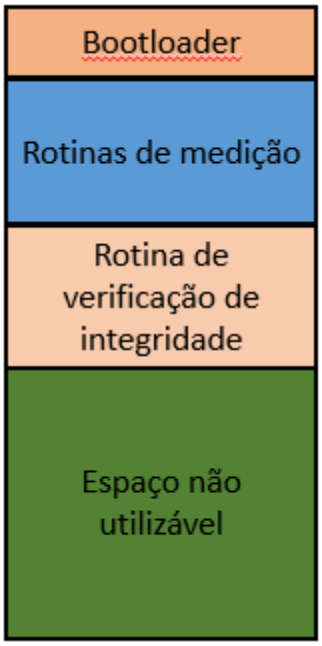
Fonte: Sinst

- Descrever o espaço de endereçamento do firmware, apresentando os valores aleatórios preenchidos ao final da área restante, se houver.
- Descrever como esses valores aleatórios foram gerados.
- Mostrar as memórias de armazenamento secundário do módulo

O requerente deve fornecer uma figura sobre a organização da memória. Todas as memórias do bloco, mesmo aquelas auxiliares (usadas temporariamente no processo de carga de software, por exemplo), estão sujeitas à verificação de integridade. O anexo A da NIT-Sinst-020 discute dois procedimentos de verificação de integridade.

O requerente deve ter especial atenção com o preenchimento das diversas memórias, devendo justificar aquelas que integralmente ou parcialmente não foram preenchidas. Para evitar um ataque que utilize o espaço de memória não ocupado deverá ser preenchido com números aleatórios, conhecidos pelo Inmetro. Dessa forma, a memória do microcontrolador terá uma organização semelhante à da figura 2. Esse preenchimento com dados aleatórios dificulta um ataque que busque alterar as rotinas de medição, pois para realizar uma alteração na rotina de medição que não seja percebida em uma auditoria futura seria necessária memória adicional. Como o espaço não utilizado de memória possui dados não desprezíveis (fazem parte do conteúdo de memória a ser auditado), eles não podem ser simplesmente sobrescritos, dificultando assim um ataque de alteração de software.

Figura 2 - Organização da memória do microcontrolador



Fonte:Sinst

Um atacante pode, entretanto, utilizar algum tipo de compressão nos dados existentes na área de memória não utilizada e realizar a descompressão, em tempo real, no instante em que a função de verificação de integridade for invocada. Assim, o atacante obterá o espaço de memória adicional, necessário para que seu ataque ocorra com êxito. Dessa forma, para minimizar o risco de um ataque de compressão deve-se preencher a área de memória não utilizada com dados aleatórios de grande dispersão. Uma sugestão para fonte de número aleatórios gerados com grande dispersão são aqueles fornecidos pelo ramdom.org. Caso seja escolhido um gerador de número pseudoaleatórios para o preenchimento da memória não ocupada, o gerador deve atender os requisitos da norma SP 800-22.

7.2.2 Identificação inequívoca do software

O item 7.2.2 elucida o requisito 3.1.2.3 da Portaria n.º 586, de 01 de novembro de 2012.

Descrição do identificador de software – deve apresentar como o identificador de software é construído, como é estruturado, e como pode ser visualizado. Deve descrever também como o identificador está indissoluvelmente associado ao software. O identificador de software é considerado um parâmetro legalmente relevante e deve estar protegido como tal.

No caso de computador tipo U, a documentação explicita um identificador para o sistema operacional e para cada componente do sistema que tenha sido modificada ou adicionada, como por exemplo, drivers. Esse identificador pode ser, por exemplo, a versão da distribuição Linux ou o número do Service Pack do Windows 7.

A documentação deve indicar no código fonte a localização das rotinas responsáveis pela criação e pelo acesso ao identificador de software (por exemplo, arquivo, classe, método, rotina, etc).

7.2.3 Verificação de integridade

O item 7.2.3 elucida o requisito 3.1.3.4 da Portaria n.º 586, de 01 de novembro de 2012.

Caso o instrumento utilize o protocolo de comunicação descrito na NIT-Sinst-020 e faça uso da ferramenta de verificação de integridade por desafio e resposta descrito no anexo A da NIT-Sinst-020, não será necessário por parte do requerente implementar uma ferramenta de verificação de integridade. Neste caso deverão ser informados:

- A especificação técnica da interface de comunicação serial de dados utilizada.
- A lista de todos os comandos gerais e específicos que podem ser ativados por meio da interface de verificação metrológica, conforme definido na Norma NIT-Sinst-020.
- A descrição detalhada dos comandos estendidos que podem ser ativados por meio da interface de verificação metrológica, apresentando as correspondentes ações passíveis de serem desencadeadas no instrumento.
- A descrição dos códigos de erro utilizados pelo requerente para informar a ocorrência de situações de erro não previstos na Norma NIT-Sinst-020 após uma solicitação de informação por parte do DVIS.
- Lista de códigos utilizados para identificação dos diferentes níveis de acesso existentes para operações de configuração, carga de software e manutenção no registro de auditoria, conforme definido na Norma NIT-Sinst-020.
- Lista de códigos utilizados para identificação dos parâmetros legalmente relevantes no registro de auditoria, conforme definido na Norma NIT-Sinst-020.
- Descrição do algoritmo de MAC utilizado durante o processo de verificação de integridade do software (Método dos Intervalos de Memória Aleatórios), conforme definido na Norma NIT-Sinst-020 anexo A.

Caso contrário, a documentação deve descrever o método de verificação de integridade do software legalmente relevante presente no instrumento e o procedimento para realiza-lo. Conforme informado no item 8.5.1 da NIT-Sinst-003, a ferramenta de verificação de integridade deve estar incluída no pacote de documentação assim como seu manual de instalação, operação e a descrição do protocolo de comunicação utilizado. Deve ser fornecida a descrição dos algoritmos e mecanismos de verificação de integridade implementados.

A documentação deve indicar no código fonte a localização das rotinas responsáveis pela verificação de integridade (por exemplo, arquivo, classe, método, rotina, etc).

7.2.4 Descrição da exatidão dos algoritmos de medição

O item 7.2.4 elucida o requisito 3.1.4.2 da Portaria n.º 586, de 01 de novembro de 2012.

Deve descrever o algoritmo através de fluxograma, destacando entradas e saídas.

Explicitar o tipo de sensor de corrente, forma de medição (dois ou quatro quadrantes) e forma de registro (registrador unidirecional, registrador bidirecional ou registrador com catraca). Relacionar claramente a forma de medição e registro na implementação, apontando as funcionalidades e os recursos computacionais utilizados como, por exemplo, se existe uma unidade de medição de energia no microcontrolador e como ela funciona. Deve atentar para os erros numéricos associados a implementação do algoritmo de medição utilizado.

A documentação deve indicar no código fonte a localização do algoritmo de medição (por exemplo, arquivo, classe, método, rotina, etc).

7.2.5 Influência da interface de entrada de dados

O item 7.2.5 elucida o requisito 3.1.5.2 da Portaria n.º 586, de 01 de novembro de 2012.

A documentação deve conter uma lista de todos os comandos, explicitando a funcionalidade de cada um deles e uma declaração explicita que todos os comandos são contemplados na lista apresentada.

Nota: Para a declaração de completude basta uma sentença clara na documentação que todos os comandos implementados no software estão descritos na documentação.

O comportamento do instrumento ao receber cada comando deve ser descrito, preferencialmente utilizando diagrama de estados e diagrama de atividades associando as correspondentes rotinas no código fonte.

Apresentar relatório de caso de testes para cada um dos comandos. Esse relatório deve conter os procedimentos utilizados para testar cada comando assim como a descrição dos ensaios realizados.

A documentação deve descrever os controles de acesso quando é possível alteração de parâmetros metrologicamente relevantes com uso de interfaces de comunicação. Preferencialmente, os algoritmos criptográficos para controle de acesso devem obedecer a norma SP 800-57 part 1.

A documentação deve indicar as rotinas do código fonte responsáveis por tratar as interfaces de dados.

7.2.6 Proteção contra mudanças acidentais/não-intencionais

O item 7.2.6 elucida o requisito 3.1.6.2 da Portaria n.º 586, de 01 de novembro de 2012. Os softwares legalmente relevantes, os parâmetros e os dados de medição devem ser protegidos contra modificações acidentais ou não intencionais. A documentação deve descrever como esse objetivo é atingido. Por exemplo: replicação dos dados com esquema de votação ou CRC dos dados.

A documentação deve apresentar como o instrumento verifica a integridade de sua memória de programa contra mudanças acidentais, dos dados de medição contra mudanças acidentais e dos parâmetros legalmente relevantes contra mudanças acidentais.

Quando as mudanças acidentais ocorrerem, a documentação deve explicitar os códigos de erro gerado e o tratamento desses erros.

A documentação deve indicar as rotinas do código fonte responsáveis por tratar a proteção contra mudanças acidentais.

7.2.7 Proteção contra mudanças intencionais não autorizadas

O item 7.2.7 elucida o requisito 3.1.7.1 da Portaria n.º 586, de 01 de novembro de 2012. O software e o hardware do instrumento devem ser projetados e construídos de tal forma que a possibilidade de seu uso impróprio ou fraudulento, quer seja intencional, não intencional ou acidental, sejam mínimas.

Devem ser descritas as medidas de proteção contra uso impróprio ou fraudulento do instrumento, incluindo meios de proteção mecânicos, eletrônicos e/ou criptográficos. Os algoritmos criptográficos utilizados devem estar de acordo com a SP 800-57 part 1.

O software e os parâmetros legalmente relevantes devem ser protegidos contra modificações inadmissíveis ou não autorizadas. A documentação deve descrever a proteção contra modificações físicas no instrumento, como por exemplo aquelas causadas pela troca indevida de unidades de memória. Caso o instrumento aceite carga de software, apenas cargas de softwares assinados pelo Inmetro devem ser autorizadas.

A documentação deve indicar as rotinas do código fonte responsáveis por tratar a proteção contra mudanças intencionais.

7.2.8 Proteção dos parâmetros

O item 7.2.8 elucida o requisito 3.1.8.3 da Portaria n.º 586, de 01 de novembro de 2012.

A documentação deve descrever cada parâmetro legalmente relevante.

Deve ser apresentado na documentação uma tabela informando, para cada parâmetro, a função no código fonte que permite sua alteração, o valor nominal, a margem de variação e local de armazenamento, semelhante a tabela 1.

Tabela 1 - Parâmetros legalmente relevantes

Parâmetro	Função	Valor nominal	Margem de variação	Local de armazenamento

Fonte: Sinst

Descrever os métodos de proteção usados para que esses sejam protegidos contra modificações não autorizadas.

Descrever os procedimentos para alteração dos parâmetros. Para medidores tarifa horo sazonal e tarifa branca, caso seja utilizado um esquema de autenticação, os requisitos de autenticação devem observar o exposto no item 8.8 da NIT-Sinst-019 que versa sobre a necessidade de validade nas portarias de aprovação de modelo para os medidores supracitados caso o esquema de autenticação adotado não seja forte o suficiente.

Descrição do procedimento de Registro de Alteração de Parâmetros Metrológicos Relevantes. Descrever o formato de todos os registros de alteração de parâmetros apontando o significado de cada campo dos registros. Se for mais apropriado, apontá-lo no momento em que o caso de teste for descrito. Os registros

não devem conter informação sigilosa, tal como a chave criptográfica usada pelos dispositivos. O registro, dependendo do evento gerado, deverá indicar quem fez o acesso, quando ocorreu tal evento e o que o evento provocou no comportamento do sistema. Descrever o mecanismo de recuperação dos registros. Identificar a capacidade de armazenamento de cada registro e temporalidade.

Descrever o procedimento de registro de alteração de parâmetros o formato dos dados armazenados.

Descrever como os parâmetros são referenciados no registro de auditoria e como podem ser visualizados.

Apresentar as garantias de como os componentes que armazenam registros de auditoria, dados e parâmetros legalmente relevantes sejam física e logicamente invioláveis.

Os registros presentes no Registro de Alteração de Parâmetros Metrológicos Relevantes devem conter, ao menos, a identificação do parâmetro, o valor antes e depois da alteração e o instante da alteração (timestamp).

7.2.9 Detecção de falha

O item 7.2.9 elucida o requisito 3.1.9.2 da Portaria n.º 586, de 01 de novembro de 2012.

Documentar a lista de falhas que são detectáveis, os respectivos algoritmos de detecção e as reações desencadeadas. O instrumento não deverá realizar medição enquanto estiver em estado de falha.

Indicar no código fonte a localização das rotinas responsáveis por detectar e tratar as falhas. Explicitar na documentação como essas rotinas são disparadas (interrupções, watchdogs, etc).

7.2.10 Validação de software

O item 7.2.10 elucida o requisito 3.1.10.1 da Portaria n.º 586, de 01 de novembro de 2012.

Descrever os casos de teste utilizados para validar o software do instrumento de medição. Montar uma tabela, conforme a tabela 2, para cada caso de teste.

Tabela 2 - 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.
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.

(Continua)

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

7.3 REQUISITOS ESPECÍFICOS DE SEGURANÇA DE SOFTWARE E HARDWARE

Os requisitos específicos tratam de aspectos técnicos referentes a funcionalidades complementares.

7.3.1 Descrição dos requisitos específicos

São considerados requisitos específicos pelo RTM anexo a Portaria n.º 586, de 01 de novembro de 2012:

- Separação das partes legalmente relevantes. Esses requisitos devem ser obedecidos quando o instrumento possuí software legalmente não relevante e software legalmente relevante. Essa abordagem permite que o requerente mantenha software legalmente não relevante no instrumento e que esse possa ser alterado sem passar por uma modificação de modelo;
- Transmissão de dados através de rede de comunicação. Quando o instrumento utiliza rede de comunicação dentro da cadeia legalmente relevante;
- Carga de software legalmente relevante. Quando é possível alterar o software legalmente relevante embarcado no Instrumento sem romper selagem principal;
- Arquiteturas baseadas em assinatura digital. Quando o sistema utiliza assinatura digital para assegurar autenticidade e irrefutabilidade.

7.3.2 Separação de software

O item 7.3.2 elucida o requisito 3.2.2.9 da Portaria n.º 586, de 01 de novembro de 2012.

Os instrumentos de medição controlados por software podem ter funcionalidades complexas e conter módulos que são legalmente relevantes e módulos que não são. A separação de software é uma metodologia que permite ao requerente modificar facilmente o software que não é legalmente relevante.

A documentação deve descrever do projeto da separação de software e/ou hardware em detalhes. O uso de diagrama de blocos corrobora na compreensão no entendimento da separação de software. Descrição e identificação dos módulos de software (programas, sub-rotinas, bibliotecas) e hardware (placas eletrônicas, componentes, transdutores) que realizem funções legalmente relevantes ou que contenham dados legalmente relevantes.

Pertencem ao software legalmente relevante, no caso de separação de baixo nível, todas as unidades de programa (sub-rotinas, procedimentos, funções, classes) e, no caso de separação de alto nível, todos os programas e bibliotecas que contribuem para 1) o cálculo de medição de valores ou que tenham um impacto sobre o mesmo e para 2) funções auxiliares como a exibição de dados, segurança de dados, armazenamento de dados, identificação de software, carga de software, transmissão ou armazenamento de dados, checagem ou armazenamentos dos dados recebidos.

Pertencem ao software legalmente relevante todas as variáveis, arquivos temporários e os parâmetros que tenham impacto sobre os valores da medição ou funções legalmente relevantes, ou ainda, dados. Caso o software legalmente relevante rode em um sistema operacional, esse sistema operacional também é legalmente relevante. Outras unidades de programa e os dados ou parâmetros não citados anteriormente compõe o software legalmente não relevante. Modificações a parte legalmente não relevante são permitidas desde que os requisitos de separação de software sejam observados.

Informações adicionais geradas pelo software que não é legalmente relevante só podem ser exibidas (ou impressas) caso elas não possam ser confundidas com as informações que se originam a partir da parte legalmente relevante.

Descrição da interface de separação, ou interface de software protetora, entre as partes legalmente relevantes e não legalmente relevantes (incluindo a descrição de funções, domínios de dados, protocolos de comunicação e barramento de dados). Quaisquer interações e fluxos de dados não devem influenciar de forma inadmissível o software legalmente relevante.

A documentação deve apresentar o protocolo de comunicação em detalhes e uma relação completa na forma de lista contendo descrição e funcionalidades de comandos de interface de separação de software e/ou hardware. Como complemento da lista de comandos especificada, deve ser apresentada uma declaração de completude dos comandos utilizados por meio das interfaces do instrumento.

Nota - a declaração de completude de comandos corresponde a uma declaração do requerente afirmando que somente os comandos descritos na documentação possuem influencia no instrumento ou sistema de medição em análise.

Caso o software legalmente relevante e o software legalmente não relevante utilizem os mesmos recursos computacionais, a documentação deve descrever como é garantida a disponibilidade necessária para a execução correta do software legalmente relevante: hierarquia de interrupção, diagrama temporal das tarefas de software, limite de tempo de execução destinado às tarefas legalmente não relevantes.

Para cada comando listado, indicar a respectiva parte do código-fonte do software que trata o comando.

7.3.3 Transmissão de dados através de rede de comunicação

O item 7.3.3 elucida o requisito 3.2.3.1 da Portaria n.º 586, de 01 de novembro de 2012.

A documentação deve descrever os mecanismos que garantem a autenticidade da informação trafegada entre os módulos. Soluções construtivas, como uso de selagem, e lógicas, como uso de ferramentas de criptografia, como por exemplo HMAC, podem ser utilizadas para garantir a autenticidade.

A documentação deve descrever os mecanismos que garantem a integridade da informação trafegada entre os módulos. Por exemplo, com uso de ferramentas de detecção de erros (CRC, checksums, funções hash criptográficas, etc) que por sua vez alteram o estado do instrumento para tratar o erro detectado. Outras abordagens, como o uso de algoritmos de correção de erro, também são possíveis.

Documentar a topologia da rede, a tecnologia utilizada para as conexões, as interfaces de rede e seus respectivos controladores, a operação de rubs, pontes, switchs, o protocolo de comunicação, etc.

A documentação deve descrever o protocolo de comunicação em detalhes. Caso seja utilizado um protocolo conhecido, como TCP/IP, a documentação deve descrever como o instrumento faz uso do protocolo para transmitir os dados.

Os mecanismos usados para descarte, ou reconstrução, dos dados corrompidos devem ser descritos na documentação, assim como a medição é protegida contra atrasos decorrentes da comunicação e os procedimentos de proteção contra a interrupção da transmissão.

Deve ser detalhado as respectivas medidas de segurança para impedir intrusão na rede interna. Soluções construtivas, como uso de selagem, e lógicas, como uso de ferramentas de autenticação forte podem ser utilizadas para proteger a rede de comunicação.

A documentação deve descrever os mecanismos para a confidencialidade de chaves criptográficas, caso essas sejam utilizadas na transmissão de dados.

7.3.4 Carga remota de software

O item 7.3.4 elucida o requisito 3.2.3.3 da Portaria n.º 586, de 01 de novembro de 2012.

A documentação deve descrever do procedimento de carga do software legalmente relevante.

A documentação deve descrever os meios pelos quais se garante que o software legalmente relevante foi avaliado e aprovado pelo Inmetro. O processo para autenticidade do software carregado remotamente passa pelo uso de uma chave pública fornecida pelo Inmetro, conforme especificado no item 10 da NIT-Sinst-003, e por uma assinatura digital do binário referente software aprovado no processo de análise de software. O memorial deve descrever qual componente do instrumento de medição verifica a assinatura digital de um software novo (assinado com a chave privada do Inmetro) a partir de uma memória auxiliar e, atestada a autenticidade, substitui o software antigo pelo novo e carrega este último durante o processo de boot.

Descrever as medidas de proteção contra carga e modificações não autorizadas do software legalmente relevante e o comportamento no caso de uma falha na carga de software e como o instrumento sinaliza o erro.

Descrever suporte de hardware. (Exemplo: existência de uma memória adicional para armazenamento temporário do software baixado).

Descrição do procedimento de registro das cargas de software e o formato dos dados. Descrever o formato de todos os registros de carga de software apontando o significado de cada campo dos registros. Se for mais apropriado, apontá-lo no momento em que o caso de teste for descrito. Os registros não devem conter informação sigilosa, tal como a chave criptográfica usada pelos dispositivos. O registro, dependendo do

evento gerado, deverá indicar quem fez o acesso, quando ocorreu tal evento e o que o evento provocou no comportamento do sistema. Descrever o mecanismo de recuperação dos registros. Identificar a capacidade de armazenamento de cada registro e temporalidade.

Descrição do procedimento de disponibilização e publicação dos registros de carga de software legalmente relevante e descrição da inviolabilidade, física e lógica, dos componentes que armazenam registros de carga de software legalmente relevante.

Para medidores tarifa horo sazonal e tarifa branca, caso seja utilizado um esquema de autenticação, os requisitos de autenticação devem observar o exposto no item 9.3 da NIT-Sinst-019 que versa sobre a necessidade de validade nas portarias de aprovação de modelo para os medidores supracitados caso o esquema de autenticação adotado não seja forte o suficiente.

Indicar no código fonte a localização das rotinas responsáveis pela carga de software.

7.3.5 Arquiteturas baseadas em assinatura digital

O item 7.3.5 elucida o requisito 3.2.5 da Portaria n.º 586, de 01 de novembro de 2012.

Medidores cujas grandezas de entrada são assinadas digitalmente só se faz necessário a entrega do código fonte que contemple todos os estados e atividades do sistema a partir da inicialização até a atividade onde ocorra a assinatura digital. Deve também ser entregue todo código fonte de interface de comunicação que causem mudança no estado do instrumento antes da assinatura digital.

A documentação deve ser explicita sobre os comandos que causam mudança no estado do instrumento antes da assinatura digital. Por exemplo, comando que alterem as grandezas de entrada, que permitam alterar a ordem da medição ou atuem no processo de assinatura digital dessas grandezas. Esses comandos não podem permitir um comportamento indevido do sistema de medição. Por exemplo, uma combinação do uso desses comandos não deve permitir a introdução de um erro sistemático em uma das grandezas de entrada.

O requerente deve prover uma ferramenta para que seja possível reconstruir a medição a partir das grandezas de entrada. A ferramenta deve permitir também avaliar a exatidão do algoritmo de medição. A exatidão do algoritmo de medição deve ser declarada na documentação.

Caso o instrumento faça carga de software, é necessário apresentar o código fonte associado as rotinas que permitem a carga e todo o software legalmente relevante deve ser assinado pelo Inmetro.

Todas as grandezas de entrada assinadas digitalmente deverão ser tratadas como parte do resultado legalmente e metrologicamente completo da medição. Deve ser provido uma forma de verificar a assinatura digital da medição.

Deve ser detalhado o gerenciamento dessas chaves que garante a sua segurança. Deve ser fornecida a especificação de segurança dos componentes que armazenam chaves criptográficas.

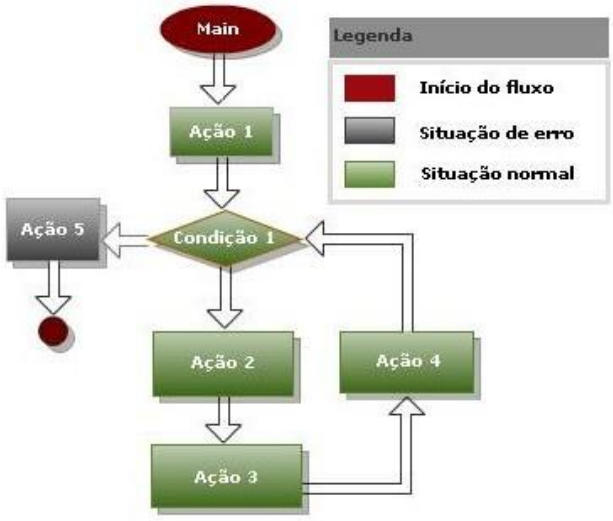
7.4 COMPORTAMENTO DINÂMICO

O item 7.4 elucida o requisito 3.3 da Portaria n.º 586, de 01 de novembro de 2012.

Descrever detalhadamente a tarefa principal do software de cada módulo utilizando um diagrama de atividades. A tarefa principal é aquela executada quando o sistema é inicializado pela função *main*. Uma tarefa é caracterizada por um fluxo único de controle sequencial dentro do software em execução (processo). Na figura 2 é apresentado um exemplo simplificado de diagrama de atividades para a tarefa main.

Descrever os módulos/bibliotecas/funções que são utilizados pela tarefa principal, descrever o estado inicial e as condições de parada (se houver) e apresentar um diagrama de tempo.

Figura 2 - Tarefa principal do módulo



Fonte: Sinst

A documentação deve apresentar casos de teste para o comportamento dinâmico do instrumento confirmando o correto funcionamento do diagrama temporal e as respectivas mudanças quando as situações de erro são provocadas.

7.5 CAPACIDADE DE PROCESSAMENTO

O item 7.5 elucida o requisito 3.4 da Portaria n.º 586, de 01 de novembro de 2012.

Documentação deve apresentar os cálculos que comprovem a capacidade de compartilhamento dos recursos de hardware que tenham uso compartilhado, tais como concentradores, redes de comunicação, switches, etc.

A documentação deve apresentar casos de teste de desempenho do instrumento, como teste de estresse, teste de carga, teste de contenção e teste de perfil de desempenho.

8 HISTÓRICO DE REVISÃO E QUADRO DE APROVAÇÃO

Revisão	Data	Itens revisados
00	Nove/2018	▪ Emissão inicial;

Quadro de Aprovação		
	Nome	**Atribuição**
Elaborado por:	Carlos Eduardo Cardoso Galhardo Fabiano O Leitão	Coordenador da qualidade do Sinst Pesquisador Tecnologista em Metrologia e Qualidade
Verificado por:	Juliana Wilm Amsterdam de J. S. M. de Mendonça	Estagiária do Sinst Coordenador da qualidade da Dimel
Aprovado por:	Bruno Erthal	Chefe do Sinst