

# ***TERASE: Template para Especificação de Requisitos de Ambiente em Sistemas Embarcados***

Luiz Eduardo Galvão Martins, Roberto de Souza Júnior,  
Hermano Peixoto de Oliveira Jr., Cecilia Sosa Arias Peixoto

*Faculdade de Ciências Exatas e da Natureza  
Universidade Metodista de Piracicaba (UNIMEP)  
Piracicaba – SP – Brasil*

{lgmartin,rsouza,cspeixot}@unimep.br,  
hermano\_peixoto@directnet.com.br

## **Abstract**

*This paper presents a template for environmental requirements specification in embedded systems, called TERASE. There is a lack of requirements specification templates that address the particularities of the embedded systems development. With TERASE embedded system engineers can improve their environmental requirements specification, as well as the communication process between hardware and software engineers. An empirical study was conducted to evaluate the template proposed. The goal of the empirical study was to apply and evaluate the results obtained with TERASE, from the point of view of the embedded system practitioner. The embedded system analyzed during the experiment was a datalogger used to capture and store temperature reading. It was adopted “in-vitro” modality for the experiment.*

## **1. Introdução**

O software de sistema embarcado tem particularidades que o distingue dos demais tipos de software. Uma vez em operação, dificilmente será possível atualizá-lo em campo devido a questões físicas ou econômicas [10]. O software tem que ser distribuído sem falhas operacionais que possam gerar responsabilidade civil e prejuízos à imagem do fabricante. Por exemplo, considere uma máquina doméstica de fazer pão, vendida em escala global por grandes multinacionais da linha branca. Não vemos atualizações de software com novas receitas distribuídas aos consumidores, nem atualizações de

software, os conhecidos *service packs*. Muitas vezes, nem é possível identificar quem são os consumidores. O que aconteceria em caso de pão queimado ou cru e alguns incêndios espalhados pelo mundo? Certamente um grave dano à marca.

Este cenário mostra a importância da Engenharia de Software como um dos elementos garantidores da qualidade do produto. O software de um sistema embarcado tem que ser sólido como uma rocha, no jargão técnico – *rock-solid*. Desenvolvedores de sistemas embarcados normalmente são engenheiros sem formação em Ciência da Computação. Isso é fácil de compreender, pois a programação de sistemas dessa natureza necessita de conhecimento especializado na plataforma de hardware do sistema e em especificações de natureza muito técnica. Por exemplo, para programar o software embarcado de um decodificador de TV digital (HDTV), o engenheiro tem que conhecer o hardware necessário e toda a norma relativa ao processo de codificação e transmissão da informação, as quais são escritas para um público com formação em engenharia elétrica. Em geral, o engenheiro de sistemas embarcados não faz uso das melhores práticas de Engenharia de Software, especialmente no que diz respeito aos requisitos e projeto do software [10]. Recentemente, praticantes da área de sistemas embarcados começaram a se movimentar em busca de métodos e técnicas da Engenharia de Software [1][3], em uma tentativa de aproveitar e adaptar o que já existe, e buscar novas técnicas que sejam aderentes às particularidades do desenvolvimento desse tipo de sistema.

Neste artigo será apresentado um *template* para especificação de requisitos de ambiente em sistemas embarcados, denominado TERASE. Este *template* tem a finalidade de facilitar a especificação dos requisitos não-funcionais que irão delinear o ambiente físico (hardware) no qual o software embarcado será implementado. Em sistemas embarcados, o software a ser implementado precisa estar totalmente alinhado com o hardware que o suportará. Os programadores deste tipo de software precisam saber exatamente quais pinos do processador serão utilizados, qual tipo de dado é tratado em cada pino, e quais os dispositivos de entrada ou saída estão associados aos pinos do processador (esses dispositivos normalmente são sensores e atuadores do sistema).

Embora existam alguns *templates* para especificação de requisitos difundidos na comunidade de Engenharia de Software, por exemplo os *templates IEEE Std 830-1998* [2] e *Volere* [6], eles são muito genéricos e não tratam adequadamente as exigências de requisitos para sistemas embarcados, principalmente no que se refere aos requisitos não-funcionais que definem o ambiente de um sistema embarcado. O *template* apresentado neste artigo foi proposto pelo *Grupo de Pesquisa em Engenharia de Software Aplicada a Sistemas Embarcados e Automação Industrial da UNIMEP*.

Um estudo experimental foi realizado com o objetivo de avaliar os benefícios e dificuldades de utilização do *template* proposto. Nesse estudo, que ocorreu dentro de um contexto real de desenvolvimento de sistemas embarcados (em uma empresa da área de eletrônica industrial da região de Campinas, Brasil), participou um engenheiro de controle e automação, que utilizou o *template* para fazer a especificação dos requisitos de ambiente de um sistema *datalogger* para captura e armazenamento de temperatura ambiente. Os resultados deste experimento serão relatados e discutidos ao longo deste artigo.

O restante deste artigo está organizado da seguinte forma: na seção 2 é apresentada uma breve caracterização de sistemas embarcados; a seção 3 apresenta o TERASE, mostrando sua estrutura e os cartões de especificação propostos; na seção 4 é explicado o formato do experimento realizado, bem como o perfil do participante do experimento, os materiais e tratamentos empíricos adotados; na seção 5 são apresentados e discutidos os resultados do experimento; e na seção 6 apresenta-se a conclusão final do trabalho.

## 2. Caracterização de Sistemas Embarcados

A definição de sistema embarcado é controversa, dada a variedade de aplicações para tais sistemas e do tipo de tecnologia empregada. É difícil conseguir uma definição geral. No entanto, existe um conjunto de características que são comuns à maioria dos sistemas embarcados, independentemente do tipo de aplicação e tecnologia subjacente. Essas características são arroladas a seguir [4][5]:

- Sistemas embarcados são sistemas computacionais que acompanham algum produto manufaturado (tipicamente um dispositivo eletromecânico) para realizar uma tarefa específica (controlada via software);
- O software é desenvolvido para um hardware específico, normalmente com alto grau de acoplamento;
- Interação fortemente com sensores e atuadores<sup>1</sup>, processando dados obtidos pelos sensores, e enviando sinais para os atuadores (que interferem com o equipamento no qual o sistema embarcado está acoplado);
- Tendem a interagir mais com sensores e atuadores do que diretamente com seres-humanos;
- O microcontrolador/processador é o componente central, que irá executar o software embarcado;
- Requisitos não-funcionais importantes: baixo consumo de energia, alta confiabilidade e alto desempenho;
- Costumam ser produzidos em grande escala (vendas na ordem de dezenas ou centenas de milhares de unidades), embarcados em um bem de consumo .

A Figura 1 apresenta uma ilustração do contexto geral dos sistemas embarcados, utilizando-se de uma visão em camadas (anéis). O anel externo representa o ambiente físico ou equipamento a ser controlado, no qual variáveis de ambiente (grandezas físicas) normalmente são monitoradas e controladas (por exemplo: temperatura, pressão, velocidade, umidade, luminosidade, torque, aceleração etc.).

---

<sup>1</sup> Um atuador é um dispositivo físico que interfere com o ambiente ou com o objeto físico controlado (exemplos: motor elétrico, válvula pneumática etc.).

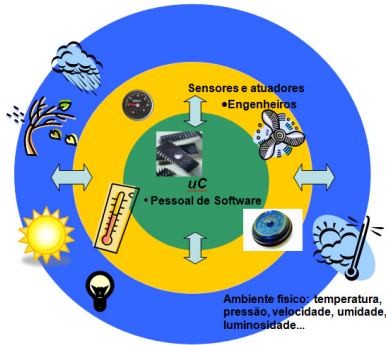


Figura 1. Contexto geral de sistemas embarcados

O anel intermediário representa os sensores e atuadores, que são utilizados como “ponte” entre o ambiente físico e o microcontrolador ( $\mu C$ ). O anel interno representa o  $\mu C$ , componente central do sistema embarcado, responsável pela execução dos softwares de monitoramento e controle do sistema.

A Figura 2 ilustra um  $\mu C$  típico utilizado em desenvolvimento de sistemas embarcados de pequeno porte. O  $\mu C$ , diferentemente dos microprocessadores de finalidade geral, integram no mesmo *chip*, além da CPU, periféricos que são muito utilizados no processamento de sinais analógicos e digitais. Periféricos que normalmente estão integrados no  $\mu C$  são: conversor A/D (análogo/digital), conversor D/A (digital/análogo), memórias ROM, RAM e EEPROM, relógio, temporizador etc.

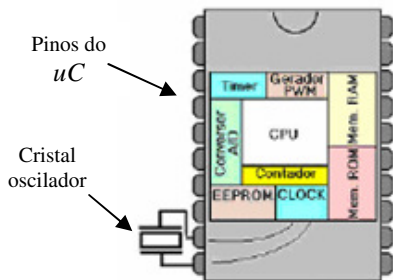


Figura 2. Esquemático de um  $\mu C$  típico [8]

### 3. TERASE: Template para Especificação de Requisitos de Ambiente em Sistemas Embarcados

O *template* proposto tem como finalidade geral oferecer uma contribuição para a melhoria da especificação de requisitos de sistemas embarcados. No entanto, o *TERASE* foi elaborado para auxiliar particularmente na especificação dos requisitos de ambiente em sistemas embarcados. Os requisitos de ambiente de um sistema embarcado envolvem dois

aspectos principais. (i) As variáveis do ambiente físico onde o sistema irá atuar (por exemplo: temperatura, pressão, força, velocidade, umidade etc.). O ambiente físico pode ser um espaço como uma sala fechada, o interior de uma caldeira, a superfície de uma máquina etc. (ii) Os dispositivos físicos que serão instalados no ambiente e interagirão com o software embarcado (direta ou indiretamente), exemplos desses dispositivos são sensores de temperatura, pressão, força etc., e os atuadores (motores elétricos, válvulas pneumáticas, inversores de frequência etc.). Os requisitos de ambiente do sistema embarcado são requisitos não-funcionais da maior importância, esses requisitos normalmente são definidos pelas equipes que projetam o hardware do sistema embarcado. As equipes de desenvolvimento de software embarcado recebem as especificações de hardware, e precisam de informações detalhadas sobre as variáveis do ambiente e os dispositivos físicos, que farão a captura e envio dos sinais controlados pelo sistema. *TERASE* funciona como um facilitador na comunicação entre as equipes de hardware e software.

Os sistemas focalizados pelo *TERASE* são aqueles construídos com processamento baseado em microcontroladores. Portanto, aplica-se a sistemas embarcados voltados à linha de eletrodomésticos e produtos de pequeno e médio porte em que um ou mais microcontroladores são utilizados para a execução do software. Sistemas embarcados microcontrolados possuem entradas e saídas (E/S) bem definidas, que são geradas por componentes eletrônicos, como sensores, teclados, memórias, relógios, botões, *leds*, *displays* e atuadores (entre outros). Os componentes de entrada enviam dados ou sinais de controle para o  $\mu C$ , e os componentes de saída recebem dados ou sinais de controle do  $\mu C$ . Uma especificação completa destes componentes, bem como as variáveis de ambiente associadas a eles, é essencial durante a especificação dos requisitos do sistema embarcado. Também é essencial a especificação cuidadosa do relacionamento entre os componentes de E/S e o  $\mu C$  adotado. Uma especificação completa dos requisitos que compõem o ambiente físico ou a planta de um sistema embarcado deve incluir:

- Variáveis do Ambiente/Planta;
- Dispositivos de Entrada;
- Dispositivos de Saída;
- Microcontrolador.

Para cada um destes itens foi elaborado um cartão de especificação de requisitos. Os cartões orientam e facilitam o registro das informações, assegurando que

os requisitos não-funcionais relacionados ao ambiente do sistema embarcado serão elicitados e especificados. Para cada “campo-conceito” do cartão, foi oferecida uma explicação de seu significado, com o intuito de facilitar o preenchimento dos cartões (apresentada a seguir).

### 3.1 Variáveis do Ambiente/Planta

Os cartões de especificação devem manter um registro detalhado dos requisitos de ambiente do sistema embarcado a ser desenvolvido. A Figura 3 apresenta o cartão de especificação para as variáveis de ambiente/planta do sistema embarcado. Cada variável de ambiente deve ser completamente especificada utilizando-se o cartão proposto. Toda variável de ambiente deve estar associada a um dispositivo físico (ex: um sensor).

Identificador#	Nome da Variável	Unidade	Tipo E ( ) S ( ) Tipo A ( ) D ( )
Conceito			Domínio
Forma de Obtenção			
<input type="checkbox"/> Medição Direta		<input type="checkbox"/> Equação	
Dispositivo Físico Associado			

Figura 3. Cartão de especificação para variável de ambiente/planta

Os campos que compõem o cartão para a especificação das variáveis do ambiente/planta são:

- **Identificador:** um rótulo numérico ou *mnemônico* que identifica a variável de forma única dentro do ambiente de sistema embarcado;
- **Nome da Variável:** um nome que tenha um significado reconhecível pelos *stakeholders* dentro do ambiente do sistema embarcado;
- **Tipo (E/S):** uma identificação se a variável será tratada como uma entrada (E) para o microcontrolador, ou uma saída (S) resultante do processamento efetuado pelo microcontrolador;
- **Tipo (D/A):** uma identificação se a variável é do tipo digital ou analógica;
- **Conceito:** uma explicação conceitual sobre a variável do ambiente/planta;
- **Unidade:** unidade de medida adotada para significar a variável, acompanhada de sua simbologia mais conhecida;
- **Domínio:** faixa de valores permitida, com os limites mínimo e máximo aceitos pelo sistema;
- **Forma de Obtenção**

- **Medição direta:** é um valor obtido diretamente do ambiente, por meio de algum dispositivo (sensor, botão etc.) que possibilite a comunicação com o microcontrolador;
- **Equação:** é um valor obtido indiretamente, que será resultado de alguma operação matemática; todas as variáveis independentes da equação devem ser especificadas;
- **Dispositivo físico associado:** nome do dispositivo físico que possibilita a interface da variável do ambiente com o microcontrolador.

### 3.2 Dispositivos de Entrada e Saída (E/S)

Para a especificação dos requisitos dos dispositivos de E/S foi criado um cartão único, que cobre os principais aspectos da especificação desses dispositivos. Dispositivos de entrada normalmente incluem teclado, sensor, botão, memória, relógio, calendário, entre outros. Dispositivos de saída normalmente incluem memória, *leds*, *displays*, atuadores entre outros dispositivos. O mesmo cartão pode ser usado para detalhar um dispositivo de entrada e de saída (eventualmente um mesmo dispositivo pode ser de entrada e saída, ex: uma memória EEPROM). A Figura 4 apresenta o cartão de especificação de dispositivo de E/S.

<input type="checkbox"/> Entrada		<input type="checkbox"/> Saída	
Identificador #	Nome do Dispositivo	Variável Associada	
Finalidade			
Tipo do sinal			
Digital		Analogico	
Quantidade de bits:		<input type="checkbox"/> Tensão ( ) Corrente ( ) Resistência	
Faixa de valores:			
Forma de Comunicação com o Microcontrolador <input type="checkbox"/> I2C ( ) SPI ( ) RS232 ( ) RS485 ( ) CAN ( ) Outra _____			Pino do microcontrolador
Resolução		Taxa de Aquisição	
Proteção			
Observações Complementares			
ENTRADA (sensor)			
Offset		Histereese	

Figura 4. Cartão de especificação para dispositivo de entrada/saída

Cada dispositivo de E/S deve ser especificado em um cartão. Os campos que compõem o cartão são:

- **Identificador:** um rótulo numérico ou *mnemônico* que identifica o dispositivo de E/S de forma única dentro do ambiente de sistema embarcado;
- **Nome do dispositivo:** uma especificação do modelo e fabricante do dispositivo;

- **Variável de E/S associada:** nome da variável do ambiente/planta associada ao dispositivo de E/S;
- **Finalidade:** uma especificação do papel deste dispositivo no contexto do sistema embarcado;
- **Tipo de sinal**
  - **Analogico:** especificação do tipo de sinal elétrico utilizado (tensão, corrente ou resistência) e a faixa de valores suportada pelo dispositivo;
  - **Digital:** especificação do número de bits fornecido pelo dispositivo;
- **Forma de comunicação com o microcontrolador:** especificação do protocolo de comunicação suportado (*I2C*, Serial, Paralelo, *CAN* etc.);
- **Pino do microcontrolador associado:** especificação do pino do microcontrolador que receberá os valores de E/S do dispositivo;
- **Taxa de aquisição:** quantidade de aquisições (leituras) feitas dentro de um intervalo de tempo;
- **Resolução:** é a sensibilidade do dispositivo, deve indicar a menor variação de intensidade da grandeza medida que pode ser detectada pelo dispositivo;
- **Proteção:** uma descrição das condições que ofereça uma proteção elétrica para os dispositivos envolvidos;
- **Offset:** é o limite máximo permitido para o desvio do valor de saída do dispositivo, deve ser descrito em valor percentual, especificar os limites acima e abaixo da curva ideal;
- **Histerese:** especificação da função que modela a histerese do dispositivo.

As variáveis de ambiente e dispositivos físicos estão vinculadas pelos campos “dispositivo físico associado” e “variável associada”, nos cartões de variável de ambiente e dispositivos físicos, respectivamente.

### 3.3 Microcontrolador

Devido a importância da especificação do microcontrolador para sistemas embarcados, um cartão foi criado especialmente para este dispositivo. Em sistemas de pequeno porte normalmente os engenheiros utilizam apenas um único *uC*. É caso de sistemas embarcados que vem embutido em eletrodomésticos. Porém, sistemas maiores podem exigir vários *uCs*, neste caso cada *uC* deve ser especificado em um cartão

(por exemplo: máquinas industriais, veículos etc.). A Figura 5 apresenta o cartão de especificação para o *uC*.

Os campos que compõem o cartão de especificação dos requisitos do *uC* estão explicados à seguir.

- **Identificador:** um rótulo numérico ou mnemônico que identifica o microcontrolador de forma única dentro do ambiente de sistema embarcado;
- **Nome do dispositivo:** uma especificação do modelo e fabricante do dispositivo;
- **Frequência de clock:** indicação da frequência de relógio em que o microcontrolador deverá operar;
- **Formas de comunicação:** descrição das possíveis formas de comunicação suportada pelo microcontrolador (*I2C*, *SPI*, Serial, Paralela, *CAN* etc.);
- **Memória de programa:** quantidade de memória de programa suportada pelo microcontrolador;
- **Memória de dados:** quantidade de memória de dados suportada pelo microcontrolador;
- **Conversor A/D:** especificação da taxa de aquisição e tamanho da palavra de dados;
- **Conversor D/A:** especificação da taxa de aquisição e tamanho da palavra de dados;
- **Associação pino/dispositivo/variável:** uma especificação do pino do microcontrolador usado, indicando o dispositivo e a variável de ambiente/planta associados ao pino; ao indicar o dispositivo físico associado ao pino do microcontrolador, deve-se considerar a necessidade de algum mecanismo extra para garantir a segurança do microcontrolador.

Identificador #	Nome do Dispositivo	Frequência de Relógio
Memória de Programa		Memória de Dados
Conversor A/D Taxa de aquisição: Tamanho da palavra (bits):		Conversor D/A Taxa de aquisição: Tamanho da palavra (bits):
Formas de Comunicação suportada pelo Microcontrolador ( ) I2C ( ) SPI ( ) RS232 ( ) RS485 ( ) USB ( ) CAN ( ) Outra _____		
Interrupções Previstas		
Associações		
Pino	Dispositivo (identificador)	Variável
Pino	Dispositivo (identificador)	Variável
Pino	Dispositivo (identificador)	Variável

Figura 5. Cartão de especificação para microcontrolador

O quadro de associações do cartão pode ser estendido para quantos pinos forem necessários. Este quadro é fundamental, pois é nele que serão identificados os pinos do *uC* que deverão ser utilizados, e esta informação posteriormente será essencial para os programadores do software embarcado.

## 4. O Experimento Realizado

Nesta seção é apresentado um relato do experimento realizado para avaliar o *template* proposto. Neste relato destaca-se o contexto em que o experimento ocorreu, os participantes envolvidos, os materiais e o tratamento experimental adotado.

### 4.1 Contexto

Neste experimento foi escolhido um sistema embarcado para ter seus requisitos de ambiente especificados, usando-se o *template* proposto. O sistema em questão foi um *datalogger*, que realiza de forma autônoma a coleta e armazenamento de leituras de temperatura em um ambiente. As leituras de temperatura ocorrem dentro de um período de tempo pré-determinado pelo usuário, por exemplo a cada 30 segundos. As leituras ficam armazenadas em uma memória EEPROM, e depois podem ser transferidas para um computador convencional.



Figura 6. Fotografia do *datalogger* especificado no experimento

O experimento foi realizado em maio de 2009, classificando-se como um estudo na modalidade *in-vitro* [7], pois o mesmo foi executado de forma controlada, por um profissional selecionado de uma

organização que atua na área de eletrônica e automação industrial da região de Campinas (SP, Brasil). A especificação técnica do *datalogger* está descrita a seguir:

*Tipo de Display*.....: LCD (1 linha x16 colunas)  
*Sensor de Temperatura* ..: LM35  
*Faixa de Temperatura*...: + 2 °C a +50 °C  
*Resolução*.....: 1 °C  
*Intervalo de Aquisição* ..: 1 a 120 min.  
*Alimentação* .....: 1 Bateria Alcalina 9V  
*Comunicação* .....: RS232  
*Status Luminoso (leds)* ..: Aquisitando/Falha  
*Microcontrolador*.....: PIC16F628

O objetivo deste experimento foi colocar em prática o *template* proposto para especificação de requisitos de ambiente de sistemas embarcados, proporcionando, a partir desta experiência, uma primeira avaliação do *template*. Os resultados do experimento irão apontar possíveis ajustes e melhorias a serem promovidas no *template*. Durante o experimento buscou-se analisar três características do *template* (que foram as variáveis dependentes observadas durante o experimento):

- **Adequação da abrangência:** verificação se as variáveis técnicas contempladas no *template* cobriam os principais componentes que constituem o ambiente físico (hardware) de um sistema embarcado;
- **Facilidade de uso:** verificação se as explicações fornecidas para o preenchimento dos cartões, bem como o *lay-out* dos mesmos, eram fáceis de compreender e utilizar;
- **Utilidade prática:** verificação do quão útil o *template* se apresentou para o engenheiro de sistema embarcado, aferindo-se o quanto a especificação dos requisitos do ambiente ficou bem documentada utilizando-se o *template*.

As hipóteses investigadas no experimento foram assim definidas:

- (a) “O *template* proposto tem abrangência adequada para a especificação dos requisitos de ambiente de sistemas embarcados”;
- (b) “O *template* proposto é fácil de utilizar por engenheiro de sistemas embarcados”;
- (c) “O *template* proposto tem utilidade prática para a especificação de requisitos de ambiente em sistemas embarcados”.

## 4.2 Participante

Neste experimento participou um engenheiro de controle e automação, com sete (7) anos de experiência em manutenção de inversores de frequência e equipamentos industriais com eletrônica embarcada, com atuação em diversas indústrias de médio e grande porte do território nacional. O engenheiro participante tinha bons conhecimentos e experiência em especificação de hardware, e estava iniciando projetos de equipamentos eletrônicos baseados em microcontroladores. O participante havia recentemente (um mês antes do experimento) terminado de projetar e implementar um *datalogger* para leitura e armazenamento de temperatura ambiente, com perspectivas, no médio prazo, de lançá-lo como produto no mercado brasileiro.

## 4.3 Materiais

Os materiais utilizados para realização do estudo experimental foram:

- Cartão de especificação de variáveis de ambiente/planta;
- Cartão de especificação de dispositivos de E/S;
- Cartão de especificação do *uC*;
- Texto de apoio explicando os campos previstos nos cartões (conforme apresentado na seção 3);
- Questionário de avaliação do *template*.

## 4.4 Tratamento Experimental

O experimento foi conduzido executando-se três etapas: (I) apresentação do *template* para o engenheiro de sistema embarcado; (II) especificação dos requisitos de ambiente do sistema *datalogger*, em que o engenheiro utilizou-se do *template* proposto; (III) avaliação do *template*, realizada pelo engenheiro de sistema embarcado participante do experimento. A execução de cada uma das etapas é comentada a seguir.

• **Etapa I: apresentação do *template* para o engenheiro de sistema embarcado.** Nesta etapa foi oferecida uma rápida explicação para o engenheiro sobre o *template*, explicando-se sua finalidade e apresentando os modelos de cartões de especificação. Foi solicitado ao engenheiro que fizesse uma leitura do descritivo de cada item do *template* (que acompanhava os cartões), e que depois ele utilizasse os cartões de especificação para documentar os requisitos de ambiente do *datalogger*. Os proponentes do *template*

não tiveram nenhum tipo de interação com o engenheiro durante uso e aplicação do *template*.

• **Etapa II: especificação dos requisitos do ambiente do sistema *datalogger* utilizando-se o *template* proposto.** Nesta etapa o engenheiro utilizou os cartões de especificação do *template*, de acordo com o entendimento que obteve advindo do descritivo dos itens do *template* e da compreensão que obteve analisando os cartões de especificação. Foram preenchidos quinze (15) cartões de especificação no total, dois (2) foram para variáveis de ambiente, doze (12) para dispositivos físicos e um (1) para o microcontrolador (vide Figura 7, 8 e 9).

• **Etapa III: avaliação do *template*, realizada pelo engenheiro de sistema embarcado participante do experimento.** Quando o engenheiro participante recebeu o *template* de especificação de requisitos, conjuntamente foi enviado um questionário de avaliação, que deveria ser respondido pelo engenheiro logo após o término da etapa de especificação. O engenheiro respondeu o questionário e o enviou para os proponentes do *template*. Cabe ressaltar que o questionário foi respondido sem a presença dos proponentes do *template*, dando maior liberdade para o respondente. Os resultados do questionário são comentados na seção 5. Depois que o engenheiro entregou o questionário para os proponentes do *template*, estes interagiram diretamente com o engenheiro, colhendo outras percepções que o engenheiro teve sobre o *template* durante o experimento realizado.

As Figuras 7, 8 e 9 apresentam alguns cartões de especificação de requisitos de ambiente, preenchidos pelo engenheiro de sistema embarcado que participou do experimento. A Figura 7 ilustra a especificação da variável de ambiente “temperatura”.

Identificador #	Nome da Variável	Unidade	Tipo_E (X) S ( )
TEMP	Temperatura	°C	
<b>Conceito</b> Medição da temperatura ambiente			<b>Domínio</b> 2°C a 50°C
<b>Forma de Obtenção</b>			
( ) Medição Direta	(X) Equação VALOR = read_adc(); delay_us(100); if (VALOR) VALOR += 1; LM35 = ((VALOR * 4) + ((int32)VALOR * 113) / 128); TEMP = ((float)LM35 / 10);		
<b>Dispositivo Físico Associado</b> Sensor de temperatura LM35D			

Figura 7. Cartão de especificação de variável de ambiente preenchido pelo engenheiro de sistema embarcado

Na Figura 8, o cartão de especificação de requisitos apresenta um dispositivo de entrada do sistema embarcado, que é um sensor de temperatura identificado como LM35D. No campo “pino do microcontrolador” é informado que o pino AN0 (que é o pino 2 do microcontrolador) será usado para receber o valor analógico das temperaturas capturadas pelo sensor. Isto indicará para o programador do software embarcado, que este é o pino que ele deverá referenciar no programa para tratar a temperatura lida pelo sensor. No campo “tipo do sinal” foi informado que este dispositivo receberá um sinal analógico, correspondente a um valor de tensão elétrica.

<input checked="" type="checkbox"/> Entrada		<input type="checkbox"/> Saída	
Identificador #	Nome do Dispositivo	Variável Associada	
LM35	Sensor LM35D	Temperatura	
Finalidade			
Leitura da temperatura ambiente			
Tipo do sinal			
Digital		Analogico	
<input type="checkbox"/> Tensão <input type="checkbox"/> Corrente <input type="checkbox"/> Resistência		<input checked="" type="checkbox"/> Tensão <input type="checkbox"/> Corrente <input type="checkbox"/> Resistência	
Quantidade de bits:		Faixa de valores:	
Forma de Comunicação com o Microcontrolador			
<input type="checkbox"/> I2C <input type="checkbox"/> SPI <input type="checkbox"/> RS232 <input type="checkbox"/> RS485 <input type="checkbox"/> CAN <input checked="" type="checkbox"/> Outra_Analógica_		Pino do microcontrolador	
Resolução		Taxa de Aquisição	
±1,5 °C			
Proteção			
Sem proteção			
Observações Complementares			
ENTRADA (sensor)			
Offset		Histerese	

Figura 8. Cartão de especificação de dispositivo de E/S preenchido pelo engenheiro de sistema embarcado

A Figura 9 apresenta o cartão de especificação de requisitos do microcontrolador adotado para o *datalogger*. O microcontrolador adotado foi da família PIC, modelo PIC16F877A (fabricado pela *Microchip Technology Inc.*), com frequência de *clock* de 4 MHz, conversor A/D de 10 bits (é o tamanho da palavra de dados que o conversor disponibilizará na memória do microcontrolador), este é um dado importante para o programador do software embarcado. O cartão também apresenta as formas de comunicação previstas que o microcontrolador utilizará no sistema embarcado (neste caso, *I2C* e *SPI* para comunicação com a memória EEPROM, e *RS232* para comunicação serial com um computador externo ao *datalogger*). Também foi informado que estava previsto o uso da interrupção de tempo TMR0 (*timer zero*), para controlar de forma precisa a taxa de aquisição das leituras de temperaturas. O quadro de associações apresentado na Figura 9 mostra a riqueza de informações que o cartão possibilitou registrar. Todas as associações do microcontrolador, indicando os pinos usados, a ligação destes com os dispositivos de E/S e as variáveis de

ambiente vinculadas, ficaram registradas no cartão. Essas informações formam um conjunto de requisitos não-funcionais do sistema embarcado, fundamental para os projetistas e programadores do software.

Identificador #	Nome do Dispositivo	Frequência de Relógio
	PIC16F877A	4MHz
Memória de Programa	Memória de Dados	
14,3kbytes	368 bytes (RAM) e 256 bytes (EPROM)	
Conversor A/D	Conversor LIA	
Taxa de aquisição:	Taxa de aquisição: n/d	
Tamanho da palavra (bits): 10 bits	Tamanho da palavra (bits): n/d	
Formas de Comunicação suportada pelo Microcontrolador		
<input checked="" type="checkbox"/> I2C <input checked="" type="checkbox"/> SPI <input checked="" type="checkbox"/> RS232 <input type="checkbox"/> RS485 <input type="checkbox"/> USB <input type="checkbox"/> CAN <input type="checkbox"/> Outra_____		
Interrupções Previstas		
Interrupção de TMR0.		
Associação		
Pino 1 (MCLR)	Dispositivo (identificador) Resistor pull-up.	Variável Reset
Pino 2 (AN0)	Dispositivo (identificador) Sensor LM35 (LM35)	Variável Temperatura
Pino 8 (RD0)	Dispositivo (identificador) Display LCD (sd_rs)	Variável Seleção Dado/Instrução
Pino 9 (RE1)	Dispositivo (identificador) Display LCD (sd_enable)	Variável Habilitação do display
Pino 13 (OSC1)	Dispositivo (identificador) Cristal Oscilador	Variável Geração do Clock
Pino 14 (OSC2)	Dispositivo (identificador) Cristal Oscilador	Variável Geração do Clock
Pino 19 (RD0)	Dispositivo (identificador) Comunicação PC Memória EEPROM e Relógio RTC (sc)	Variável Pino de clock

Figura 9. Cartão de especificação do microcontrolador preenchido pelo engenheiro de sistema embarcado

## 5 Discussão dos Resultados

Os resultados do experimento foram analisados a partir de dois pontos de vistas. O primeiro ponto de vista foi o do engenheiro de sistema embarcado que participou do experimento, e respondeu ao questionário de avaliação que lhe foi passado. O questionário consistia de sete (7) perguntas simples e diretas sobre a experiência de uso do *template* (vide Quadro 1). O segundo ponto de vista foi o dos proponentes do *template*, que fizeram uma análise detalhada dos cartões de especificação preenchidos pelo engenheiro de sistema embarcado.

### 5.1 Análise dos Resultados do Experimento

Do ponto de vista do engenheiro evidenciou-se uma avaliação positiva quanto ao uso do *template*. Pelas



respostas do questionário de avaliação, ficou claro que o usuário do *template* considerou o mesmo com boa cobertura técnica, fácil de usar e muito útil como instrumento de especificação de requisitos de ambiente (vide Quadro 1). No entanto, do ponto de vista dos proponentes do *template*, uma série de considerações sobre ajustes e melhorias foram apontadas. Dividiremos esta discussão em duas partes, às quais denominaremos: evidências positivas quanto ao uso do *template*; e evidências negativas quanto ao uso do *template*.

### Evidências Positivas quanto ao Uso do *Template* (pontos fortes):

- ✓ O *template* demonstrou-se de fácil utilização, o experimento evidenciou que todos os modelos de ficha de especificação foram utilizados sem dificuldades;
- ✓ A abrangência do *template* demonstrou-se suficiente, pois no experimento não houve registro de ausência de item relevante no *template*;
- ✓ A estrutura proposta para o *template*, dividindo-o em três cartões de especificação (variáveis de ambiente, dispositivos de entrada/saída, e microcontrolador), se mostrou adequada, pois no experimento não houve registro de inconveniência, ausência ou confusão quanto ao uso dos cartões propostos;
- ✓ O relacionamento entre as fichas de especificação (variável -> dispositivo de E/S -> Microcontrolador) se mostrou adequado, pois todos os campos que estabeleciam relacionamentos entre os cartões foram preenchidos adequadamente pelo engenheiro de sistemas embarcados; ademais não houve registro de confusão ou mal-entendido quanto ao uso desses campos de relacionamentos.

Quadro 1 - Resultado do questionário respondido pelo engenheiro de sistema embarcado

1) O *template* proposto atendeu às necessidades de especificação dos requisitos de ambiente do seu sistema embarcado de forma:  
 Completa     Parcial     Incompleta

2) Se você respondeu parcial na pergunta anterior, indique o que não foi atendido pelo *template*:

3) Indique possíveis melhorias no *template* (se entender conveniente):

4) O *template* foi fácil de usar?  
 Sim     Não  
 Se não, quais foram as dificuldades?

5) Você avalia o grau de utilidade do *template* como:  
 Muito útil     Útil     Pouco útil     Desnecessário

6) Você adotaria este *template* no desenvolvimento de seus sistemas?  
 Sim     Não    Por quê?

7) Se achar conveniente acrescente comentários sobre o *template*.  
 "Muito útil, principalmente a Ficha de Especificação do Microcontrolador, que dá uma boa visão de como será o hardware, facilitando a geração do esquema eletrônico do produto, tornando-se mais um documento sobre o projeto."

### Evidências Negativas quanto ao Uso do *Template* (pontos fracos, que merecem ajustes futuros):

- ✓ O *datalogger* que foi utilizado como sistema embarcado, e teve os requisitos de ambientes especificados, apresentava 8 variáveis de ambiente (uma variável analógica [temperatura] e sete variáveis binárias [5 botões de comandos, e 2 *leds* indicadores de estado]), no entanto o engenheiro considerou como variável de ambiente apenas a temperatura; **Recomendação:** verificar o texto explicativo do *template*, e explicitar a necessidade da especificação das variáveis binárias (digitais) do ambiente.
- ✓ O projetista tratou a temperatura como sendo duas variáveis diferentes, uma para a aquisição da temperatura (variável de entrada) e outra para o envio a um computador externo, via porta serial RS232 (variável de saída), sendo que de fato a temperatura era apenas uma única variável de ambiente (variável de entrada); aparentemente houve confusão entre variável de ambiente e dispositivo de entrada ou saída; **Recomendação:** explicitar no texto explicativo do *template* que o conceito de variável de ambiente independe dos dispositivos de entrada ou saída; explicitar que uma mesma variável pode, eventualmente, estar associada a um dispositivo de entrada e de saída.
- ✓ No preenchimento dos cartões de especificação dos dispositivos de entrada e saída, alguns campos não foram preenchidos pelo engenheiro; **Recomendação:** diferenciar e indicar quais são os campos opcionais e obrigatórios do *template*.
- ✓ No preenchimento das fichas de especificação dos dispositivos de entrada e saída, para alguns dispositivos a comunicação utilizada foi do tipo paralela, que não estava prevista no campo "formas de comunicação com o microcontrolador"; **Recomendação:** adicionar o tipo "paralela" como uma das formas de comunicação previstas com o microcontrolador.
- ✓ No preenchimento das fichas de especificação dos dispositivos de entrada ou saída, há um campo para indicar o pino do microcontrolador ao qual o dispositivo associa-se, porém para alguns dispositivos mais de um pino estavam associados (exemplo: *display LCD*, memória EEPROM, RTC e porta serial); **Recomendação:** prever no texto explicativo do *template* e nos cartões de especificação, que mais de um pino do

microcontrolador pode estar associado ao dispositivo de entrada e saída.

- ✓ Em alguns dispositivos de entrada e saída, houve confusão quanto a indicação da variável de ambiente associada (por exemplo *display LCD*, memória EEPROM); essa confusão está associada ao fato de que alguns dispositivos, como os exemplificados, podem não ter uma única variável de ambiente associada; **Recomendação:** explicitar no texto explicativo do *template* que um dispositivo de entrada ou saída pode ter mais que uma variável de ambiente associada; o conjunto de variáveis deve ser obrigatoriamente indicado no campo “variável associada”.
- ✓ Sobre a especificação dos botões presentes na interface com o usuário, que estariam associados às variáveis de entrada, no campo “variável associada” o engenheiro anotou “I/O” (isto ocorreu para todos os botões); no campo “nome do dispositivo” o engenheiro tratou cada botão como sendo teclado; **Recomendação:** explicitar no texto explicativo do *template* que no campo “variável associada” do cartão de especificação de dispositivos de entrada ou saída, deve-se anotar o identificador correspondente utilizado no cartão de especificação de variável do ambiente; disponibilizar no cartão de especificação de dispositivos de entrada ou saída os tipos de dispositivos de E/S mais utilizados em sistemas embarcados;
- ✓ Sobre a especificação dos LEDs presentes na interface com o usuário, que estariam associados às variáveis de saída, no campo “variável associada” o engenheiro anotou “I/O” (isto ocorreu para todos os LEDs); no campo “nome do dispositivo” o engenheiro tratou cada LED como sendo um *status*; **Recomendação:** idem à anterior.
- ✓ O engenheiro, por iniciativa própria, apresentou a lista de relacionamentos “Pino do microcontrolador - Dispositivos de E/S – Variável” em ordem crescente pelo número do pino do microcontrolador, e ao lado do número anotou o mnemônico correspondente (de acordo com o *datasheet* do microcontrolador, por exemplo: RB2, RD1, AN0 etc.); **Recomendação:** explicitar no texto explicativo do *template*, que na ficha de especificação do microcontrolador, os relacionamentos envolvendo “Pino – Dispositivo – Variável” devem ser apresentados em ordem crescente pelo número dos pinos (pois isto facilita encontrar um relacionamento de um pino do microcontrolador especificado no cartão); explicitar

a necessidade de identificar os pinos do microcontrolador pelo número e pelo mnemônico indicado no *datasheet*.

## 5.2 Integração do TERASE com outros Templates

TERASE focaliza um aspecto específico do domínio de sistemas embarcados, que são os requisitos de ambiente físico do sistema. *Templates* como o *VOLERE* e o *IEEE STD 830-1998* indicam a necessidade da captura de vários requisitos não-funcionais, entre eles os requisitos operacionais e ambientais do sistema, e as características físicas do produto [2][6]. No entanto, esses *templates* não instrumentalizam o engenheiro de requisitos com modelos específicos para a captura desses requisitos não-funcionais (e de fato este não é o propósito desses *templates*).

Assim, TERASE pode ser utilizado como um instrumento particular para a especificação de requisitos de ambiente em sistemas embarcados, dentro da estrutura geral oferecida, por exemplo, pelo *template Volere*.

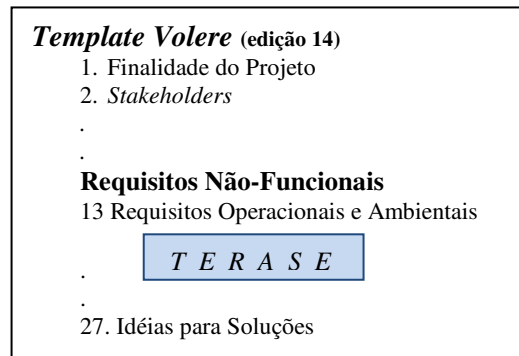


Figura 10. Integração do TERASE com o *template Volere*

A Figura 10 ilustra em que seção do *template Volere* o TERASE pode ser agregado, integrando-se a um *template* genérico e oferecendo os recursos apropriados para a especificação dos requisitos ambientais de sistemas embarcados. Dessa forma, engenheiros de sistemas embarcados contarão com um instrumento mais adequado para produzir uma especificação de requisitos mais completa de seus sistemas. TERASE pode facilmente ser integrado a *templates* de finalidade geral, enriquecendo-os na seção que trata dos requisitos não-funcionais.

### 5.3 Conformidade do *TERASE* em Relação à Recomendação *IEEE Std 830-1998*

O *IEEE Std 830-1998* é um documento que apresenta um conjunto de recomendações para a criação e manutenção de uma boa especificação de requisitos de software. De acordo com as recomendações do *IEEE Std 830-1998*, uma boa especificação de requisitos de software deve ser:

- Correta;
- Não-ambígua;
- Completa;
- Consistente;
- Classificada por importância;
- Verificável;
- Modificável;
- Rastreável.

A proposta de *TERASE* é ser um *template* para especificação de requisitos de ambiente em sistemas embarcados, mas que deve ser utilizado em conjunto com outros *templates* de especificação de requisitos. *TERASE* não endereça a especificação de requisitos funcionais, mas sim requisitos não-funcionais para sistemas embarcados, particularmente os requisitos do ambiente físico do sistema (constituído por sensores, atuadores, componentes de interface com o usuário, e a interface destes componentes com o microcontrolador do sistema).

Assim, *TERASE* auxilia na criação de uma boa especificação de requisitos de sistemas embarcados, contribuindo principalmente em relação aos atributos não-ambigüidade, completude, consistência, verificação e rastreabilidade. *TERASE* auxilia para que a especificação torne-se não-ambígua na medida em que “força” uma especificação detalhada dos componentes físicos do sistema (principalmente quanto à especificação das características técnicas dos componentes). Auxilia na completude, pois leva o especificador a identificar todas as variáveis de ambiente do sistema, bem como todos os componentes de entrada, saída e processamento do sistema embarcado (e suas interfaces de comunicação). Contribui para aumentar a consistência interna da especificação de requisitos, pois os cartões de especificação exigem um alto nível de detalhamento das informações, fazendo com que possíveis conflitos de especificação tornem-se extremamente reduzidos.

*TERASE* ainda contribui para tornar a especificação de requisitos verificável. O caráter quantitativo exigido

em alguns campos dos cartões de especificação, como *resolução*, *taxa de aquisição*, *offset* e *histerese*, contribuem de forma significativa para a verificação do atendimento dos requisitos relacionados aos componentes de entrada e saída do sistema embarcado. *TERASE* também contribui para um tipo especial de rastreabilidade, característico em sistemas embarcados, que é o rastreamento das variáveis de ambiente do sistema até os pinos de *IO* do microcontrolador, passando pelos componentes intermediários do sistema (por exemplo: sensores, botões, memórias, atuadores etc.). Os cartões de especificação possuem campos que garantem este tipo especial de rastreabilidade.

É importante lembrar que em sistemas embarcados os clientes de um produto, na maioria das vezes, são engenheiros ou técnicos que conhecem bem as características técnicas das variáveis envolvidas no sistema, e que utilizam um linguajar técnico para informar os requisitos do sistema, por exemplo: *resolução* e *taxa de aquisição* de um sensor; características de consumo de energia de um dispositivo *display*; características de sinais elétricos de um motor (*tensão*, *corrente* e *resistência*) etc. *TERASE* foi concebido levando-se em conta este perfil técnico dos *stakeholders* em sistemas embarcados.

## 6. Conclusão

Uma das grandes dificuldades enfrentadas atualmente no desenvolvimento de sistemas embarcados, está relacionada a problemas de comunicação entre as equipes de hardware e software envolvidas na construção de tais sistemas [1][3]. Os engenheiros de software precisam receber uma especificação precisa do ambiente físico no qual o software embarcado será implantado. Essa especificação é feita pelos engenheiros de hardware. Se houver falha no processo de comunicação entre as equipes de hardware e software, certamente isto trará impactos no cronograma e no custo final de desenvolvimento do sistema embarcado. *TERASE* é um instrumento que pode facilitar a especificação de requisitos de ambiente em sistemas embarcados, e contribuir para tornar mais confiável a comunicação entre as equipes de hardware e software alocadas no desenvolvimento de sistemas embarcados. Embora existam alguns *templates* de especificação de requisitos bem definidos que auxiliam as equipes de desenvolvimento na documentação dos requisitos do sistema, por exemplo, os *templates* sugeridos em *IEEE STD 830-1998* e o *template Volere*, esses *templates* são muito genéricos e não atendem adequadamente as

especificidades das equipes que desenvolvem sistemas embarcados.

Neste artigo apresentamos o *template TERASE*, que é uma proposta para atender as necessidades dos engenheiros de sistemas embarcados no que se refere à especificação dos requisitos do ambiente físico (hardware) do sistema embarcado. Este *template* integra-se facilmente aos *templates* de finalidade geral.

O experimento reportado neste artigo apresentou resultados promissores sobre o *template* de especificação de requisitos proposto. As hipóteses investigadas foram confirmadas pelo engenheiro de sistema embarcado participante do experimento, embora a equipe proponente do *template* tenha identificado uma série de ajustes que precisam ser promovidos no mesmo. O experimento na modalidade *in-vitro* demonstrou-se adequado para o tipo de aplicação e avaliação desejadas, devendo ser mantido para os próximos estudos empíricos a serem realizados.

Os resultados obtidos com o experimento realizado apresentam indícios dos benefícios que os engenheiros de sistemas embarcados podem conseguir adotando um *template* de especificação de requisitos aderente às particularidades dos sistemas embarcados. Os resultados positivos oriundos desse experimento têm estimulado o grupo de pesquisa responsável pelo *TERASE* a revisar e expandir o mesmo. O *template* apresentado está sendo expandido para cobrir também os requisitos de rede de comunicação em sistemas embarcados distribuídos, característicos da área de sistemas embarcados automotivos [9][11], que é um nicho de mercado com forte demanda de desenvolvimento de software embarcado.

*TERASE* apresenta indícios de que pode oferecer uma contribuição importante para a etapa de especificação de requisitos em sistemas embarcados. Com isso, diminuirá a escassez de recursos específicos da Engenharia de Requisitos voltados para esse tipo de sistemas.

## 7. Referências

- [1] Graaf, B.; Lormans, M. and Toeteneel, H. "Embedded Software Engineering: The State of the Practice", IEEE Software, 2003.
- [2] IEEE Std 830-1998 *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Computer Society, 1998.
- [3] Liggesmeyer, P. and Trapp, M. "Trends in Embedded Software Engineering", IEEE Software, May/June 2009. pp. 19 – 25.
- [4] Noergaard, T. *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*, Elsevier, 2005.
- [5] Oliveira, A. S. e Andrade, F. S. *Sistemas Embarcados: Hardware e Firmware na Prática*, Editora Érica, 2006.
- [6] Robertson, J. and Robertson, S. *Volere Requirements Specification Template*, Edition 14, January/2009. Disponível em <http://www.volere.co.uk>.
- [7] Travassos, G.H. and Barros, M.O. "Contributions of In Vitro and In Silico Experiments for the Future of Empirical Studies in Software Engineering", Workshop Series on Empirical Software Engineering, WSESE, ESERNET, 2003.
- [8] Corteletti, D. *Introdução à Programação de Microcontroladores Microchip PIC*, Dossiê Técnico, SENAI-RS, Centro Tecnológico de Mecatrônica, 2006.
- [9] Weber M. and Weisbrod, J. "Requirements Engineering in Automotive Development: Experiences and Challenges". IEEE Software, January/February 2003. p.16-24.
- [10] Broy, M. "Requirements Engineering for Embedded Systems", (FEmSys'97) Workshop on Formal Design of Safety Critical Embedded Systems, Germany, 1997.
- [11] Pretschner, A.; Broy, M.; Krüger, I. H. and Stauner, T. *Software Engineering for Automotive Systems: A Roadmap. Proceedings of the Future of Software Engineering (FOSE'07)*, IEEE Computer Society, 2007.