

# **Uma aplicação *Web* para apoio a laboratórios remotos reconfiguráveis baseados na norma IEEE 1451.0**

**Diogo Carlos Almeida Azevedo**

**Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Arquiteturas, Sistemas e Redes**

**Orientador: Professor Doutor Ricardo Jorge Guedes da Silva Nunes da Costa**

**Júri:**

Presidente:

Professor Doutor Luis Miguel Moreira Lino Ferreira, do Departamento de Engenharia Informática do Instituto Superior de Engenharia do Porto

Vogais:

Professor Doutor Gustavo Ribeiro Da Costa Alves, do Departamento de Engenharia Electrotécnica do Instituto Superior de Engenharia do Porto

Professor Doutor Ricardo Jorge Guedes da Silva Nunes da Costa, do Departamento de Engenharia Electrotécnica do Instituto Superior de Engenharia do Porto



# Dedicatória

*Aos meus pais e à Ana.*



# Resumo

Nos últimos anos, o processo de ensino e aprendizagem tem sofrido significativas alterações graças ao aparecimento da *Internet*. Novas ferramentas para apoio ao ensino têm surgido, nas quais se destacam os laboratórios remotos. Atualmente, muitas instituições de ensino disponibilizam laboratórios remotos nos seus cursos, que permitem, a professores e alunos, a realização de experiências reais através da *Internet*. Estes são implementados por diferentes arquiteturas e infraestruturas, suportados por vários módulos de laboratório acessíveis remotamente (e.g. instrumentos de medição). No entanto, a sua inclusão no ensino é ainda deficitária, devido: i) à falta de meios e competências técnicas das instituições de ensino para os desenvolverem, ii) à dificuldade na partilha dos módulos de laboratório por diferentes infraestruturas e, iii) à reduzida capacidade de os reconfigurar com esses módulos.

Para ultrapassar estas limitações, foi idealizado e desenvolvido no âmbito de um trabalho de doutoramento [1] um protótipo, cuja arquitetura é baseada na norma IEEE 1451.0 e na tecnologia de FPGAs. Para além de garantir o desenvolvimento e o acesso de forma normalizada a um laboratório remoto, este protótipo promove ainda a partilha de módulos de laboratório por diferentes infraestruturas. Nesse trabalho explorou-se a capacidade de reconfiguração de FPGAs para embutir na infraestrutura do laboratório vários módulos, todos descritos em ficheiros, utilizando linguagens de descrição de *hardware* estruturados de acordo com a norma IEEE 1451.0. A definição desses módulos obriga à criação de estruturas de dados binárias (*Transducer Electronic Data Sheets*, TEDSs), bem como de outros ficheiros que possibilitam a sua interligação com a infraestrutura do laboratório. No entanto, a criação destes ficheiros é bastante complexa, uma vez que exige a realização de vários cálculos e conversões. Tendo em consideração essa mesma complexidade, esta dissertação descreve o desenvolvimento de uma aplicação *Web* para leitura e escrita dos TEDSs. Para além de um estudo sobre os laboratórios remotos, é efetuada uma descrição da norma IEEE 1451.0, com particular atenção para a sua arquitetura e para a estrutura dos diferentes TEDSs. Com o objetivo de enquadrar a aplicação desenvolvida, efetua-se ainda uma breve apresentação de um protótipo de um laboratório remoto reconfigurável, cuja reconfiguração é apoiada por esta aplicação. Por fim, é descrita a verificação da aplicação *Web*, de forma a tirar conclusões sobre o seu contributo para a simplificação dessa reconfiguração.

**Palavras-chave:** Experimentação remota, norma IEEE 1451.0, TEDSs, reconfiguração de laboratórios remotos.



# Abstract

In recent years, the teaching and learning processes have changed significantly, due to the emergence of the Internet. New tools emerged, namely the so-called remote laboratories (also named weblabs). Some educational institutions are now providing weblabs in their courses, which allow teachers and students to conduct real experiments over the Internet. These are implemented by different architectures and infrastructures, supported by several weblab modules remotely accessible (e.g. measuring instruments). However, the inclusion of weblabs in education is still reduced due to: i) the lack of resources and technical skills of some institutions for their development, ii) the difficulty to share the weblab modules with different infrastructures and, iii) the reduced capacity to reconfigure them with those modules.

To overcome these limitations, it was designed and developed, within the scope of a PhD thesis [1], a weblab prototype following an architecture based on the IEEE 1451.0 Std. and FPGAs technology. Besides ensuring the standardized weblab development and access, the prototype also enables sharing their weblab modules by different infrastructures. This work explored the reconfiguration capability of FPGAs to embed several modules in weblab infrastructures, all described in files using hardware description languages and structured according to IEEE 1451.0 Std. The definition of these modules requires the creation of binary data structures named Transducer Electronic Data Sheet (TEDSs), as well as other files to enable their interconnection with the weblab infrastructure. However, the creation of these files is quite complex, since it requires performing several calculations and conversions. Considering this same complexity, this dissertation describes the development of an application for reading and writing the TEDSs files. Besides the study of weblabs, the IEEE 1451.0 Std. is presented, with particular attention to its architecture and the TEDSs' structures. To contextualize the developed Web application, a brief description of a reconfigurable weblab prototype is presented, whose reconfiguration is supported by this application. The work described in this dissertation also demanded the perception of the functioning of the existing prototype of a reconfigurable weblab, with the purpose to frame the developed application. Finally, the experimental verification of the Web application is described in order to present conclusions about their ultimate contribution in the reconfiguration of that weblab.

**Keywords:** Remote experimentation, IEEE 1451.0 Std., TEDSs, reconfigurable remote laboratories, weblabs.





# Agradecimentos

Dedico este espaço a todas as pessoas que, de certa forma, contribuíram para o desenvolvimento e concretização deste projeto.

Deixo um agradecimento especial ao Professor Doutor Ricardo Costa na qualidade de meu orientador. Agradeço-lhe pela total disponibilidade, compreensão e paciência que teve na orientação do trabalho e na escrita desta dissertação.

Destaco um profundo agradecimento aos meus pais, que são a base de tudo o que sou hoje e a quem só tenho a agradecer, bem como à Ana, pela importância que tem na minha vida.

Deixo igualmente uma palavra de agradecimento a todos os restantes familiares que sempre acreditaram no meu esforço, principalmente aos meus avós.

Quero agradecer ainda aos amigos que me acompanham, bem como a todos aqueles que sempre me deram uma palavra de incentivo durante a realização deste projeto. Agradeço igualmente ao movimento Jovens Sem Fronteiras que me tornaram numa pessoa melhor.



# Índice

<b>1</b>	<b>Introdução .....</b>	<b>1</b>
1.1	Enquadramento e motivação .....	1
1.2	Objetivos e principais contribuições .....	2
1.3	Considerações de escrita.....	3
1.4	Organização do documento.....	3
<b>2</b>	<b>Experimentação remota no ensino .....</b>	<b>5</b>
2.1	O ensino à distância .....	5
2.2	Os laboratórios remotos no apoio ao ensino .....	7
2.2.1	Atores envolvidos.....	8
2.2.2	Vantagens vs desvantagens .....	8
2.2.3	Ferramentas de comunicação adotadas .....	10
2.3	Arquitetura de um laboratório remoto .....	10
2.3.1	Camada lógica .....	12
2.3.2	Camada física .....	13
2.4	Exemplos de laboratórios remotos .....	15
2.5	Problemas atuais e solução proposta .....	18
2.6	Sumário .....	19
<b>3</b>	<b>A norma IEEE 1451.0 no desenvolvimento de laboratórios remotos .....</b>	<b>21</b>
3.1	A família de normas IEEE 1451.x.....	21
3.2	A norma IEEE 1451.0: arquitetura e funcionalidades.....	24
3.3	Estrutura de dados: TEDS .....	25
3.3.1	Vantagens dos TEDSs.....	27
3.3.2	Estrutura de um TEDS .....	27
3.4	Exemplo de um laboratório remoto reconfigurável baseado na norma IEEE 1451.0.....	29
3.4.1	Arquitetura adotada .....	30
3.4.2	Aspetos de desenvolvimento .....	31
3.4.3	Funcionalidades e interface da ferramenta de reconfiguração.....	32
3.5	Sumário .....	34
<b>4</b>	<b>Descrição da aplicação desenvolvida.....</b>	<b>35</b>
4.1	Motivação subjacente ao desenvolvimento .....	35
4.2	Arquitetura.....	37
4.3	Funcionalidades .....	38
4.3.1	Ferramentas.....	39
4.3.2	Interfaces.....	41
4.3.3	Sequência de utilização .....	43

4.4	Aspetos de desenvolvimento .....	45
4.4.1	Software utilizado .....	45
4.4.2	Base de dados.....	46
4.5	Sumário .....	47
<b>5</b>	<b>Verificação experimental da aplicação.....</b>	<b>49</b>
5.1	Objetivos e estratégia .....	49
5.2	Cenário adotado .....	50
5.3	Especificação de ficheiros .....	51
5.3.1	Módulo de entrada .....	53
5.3.2	Módulo de saída .....	60
5.3.3	Módulo SMCM.....	64
5.3.4	Módulo IEEE1451.0 .....	69
5.3.5	Ficheiro <i>Map Table</i> .....	73
5.3.6	Ficheiro de <i>Configuração</i> .....	77
5.4	Avaliação qualitativa .....	81
5.5	Sumário .....	82
<b>6</b>	<b>Conclusão.....</b>	<b>83</b>
6.1	Resumo.....	83
6.2	Principais dificuldades e limitações.....	84
6.3	Perspetivas de desenvolvimento.....	85
	<b>Referências .....</b>	<b>87</b>
	<b>Anexo A.....</b>	<b>89</b>
	<b>Anexo B.....</b>	<b>97</b>
	<b>Anexo C.....</b>	<b>103</b>
	<b>Anexo D.....</b>	<b>107</b>

# Lista de Figuras

Figura 2.1: Tipos de ambientes laboratoriais.....	7
Figura 2.2: Arquitetura cliente-servidor utilizada no acesso a uma infraestrutura laboratorial remota.....	11
Figura 2.3: Camada lógica e camada física de uma infraestrutura laboratorial remota. ....	12
Figura 2.4: Controlo síncrono e assíncrono de uma infraestrutura laboratorial remota. ....	13
Figura 2.5: Interligação de equipamentos numa infraestrutura laboratorial remota. ....	14
Figura 2.6: Sistema de comutação entre várias experiências numa infraestrutura laboratorial remota. ....	15
Figura 3.1: Modelo de um transdutor inteligente. ....	22
Figura 3.2: Arquitetura da norma IEEE 1451.0.....	24
Figura 3.3: Capacidade <i>plug-and-play</i> dos módulos dos transdutores.....	25
Figura 3.4: Arquitetura do protótipo do laboratório remoto reconfigurável. ....	30
Figura 3.5: Diagrama de blocos da arquitetura do laboratório remoto. ....	32
Figura 3.6: Interface da ferramenta de reconfiguração de laboratórios remotos. ....	33
Figura 4.1: Arquitetura da aplicação desenvolvida.....	38
Figura 4.2: Página inicial da aplicação.....	41
Figura 4.3: Interface da ferramenta que permite a escrita, leitura ou alteração do <i>End User Application Specific TEDS</i> . ....	43
Figura 4.4: Exemplos do registo de atividade do <i>Build</i> . ....	44
Figura 4.5: Painel com os dados criados do <i>End User Application Specific TEDS</i> .....	44
Figura 4.6: Painel com a hiperligação para o ficheiro criado. ....	45
Figura 4.7: Exemplo do ficheiro XML para a ferramenta <i>Config Map Table</i> . ....	47
Figura 5.1: Configurações do laboratório remoto. ....	51
Figura 5.2: Diagrama geral da utilização da aplicação desenvolvida. ....	52
Figura 5.3: Interface da ferramenta <i>User's Transducer Name TEDS</i> – identificação do TEDS. ....	53
Figura 5.4: Interface da ferramenta <i>User's Transducer Name TEDS</i> – formato do TEDS. ....	54
Figura 5.5: Interface da ferramenta <i>User's Transducer Name TEDS</i> – leitura do TEDS. ....	55
Figura 5.6: Interface da ferramenta <i>User's Transducer Name TEDS</i> – bloco de dados. ....	55
Figura 5.7: Interface da ferramenta <i>TransducerChannel TEDS</i> – <i>calibration key</i> e <i>TC type key</i> . ....	56
Figura 5.8: Interface da ferramenta <i>TransducerChannel TEDS</i> – <i>physical units</i> . ....	57
Figura 5.9: Interface da ferramenta <i>TransducerChannel TEDS</i> – <i>operational lower range limit</i> , <i>operational upper range limit</i> e <i>worst-case uncertainty</i> . ....	57
Figura 5.10: Interface da ferramenta <i>TransducerChannel TEDS</i> – <i>self-test key</i> .....	57

Figura 5.11: Interface da ferramenta <i>TransducerChannel</i> TEDS – <i>sample definition</i> . .....	58
Figura 5.12: Interface da ferramenta <i>TransducerChannel</i> TEDS – <i>data set definition</i> . .....	58
Figura 5.13: Interface da ferramenta <i>TransducerChannel</i> TEDS – <i>timing-related information</i> . .	59
Figura 5.14: Interface da ferramenta <i>TransducerChannel</i> TEDS – <i>sampling attribute</i> . .....	59
Figura 5.15: Interface da ferramenta <i>Manufacturer-defined</i> TEDS. ....	65
Figura 5.16: Interface da ferramenta <i>Manufacturer-defined</i> TEDS – <i>free data block</i> . ....	66
Figura 5.17: Interface da ferramenta <i>Meta-</i> TEDS – <i>UUID</i> . .....	70
Figura 5.18: Interface da ferramenta <i>Meta-</i> TEDS – <i>timing-related information</i> . ....	71
Figura 5.19: Interface da ferramenta <i>Config Map Table</i> - <i>TIM</i> . ....	75
Figura 5.20: Interface da ferramenta <i>Config Map Table</i> – <i>TransducerChannels</i> . ....	76
Figura 5.21: Interface da ferramenta <i>Config Map Table</i> – leitura do ficheiro. ....	76
Figura 5.22: Interface da ferramenta <i>Config Map Table</i> – opções de alteração. ....	77
Figura 5.23: Interface da ferramenta <i>Configure Weblab Modules Connections</i> – <i>TIM</i> . .....	78
Figura 5.24: Interface da ferramenta <i>Configure Weblab Modules Connections</i> – módulo de laboratório. ....	79
Figura 5.25: Interface da ferramenta <i>Configure Weblab Modules Connections</i> – leitura do ficheiro. ....	80
Figura 5.26: Interface da ferramenta <i>Configure Weblab Modules Connections</i> – opções de alteração. ....	80
Figura B.1: Interface da ferramenta <i>Meta-</i> TEDS. ....	97
Figura B.2: Interface da ferramenta <i>TransducerChannel</i> TEDS. ....	98
Figura B.3: Interface da ferramenta <i>User’s Transducer Name</i> TEDS. ....	99
Figura B.4: Interface da ferramenta <i>End User Application Specific</i> TEDS. ....	99
Figura B.5: Interface da ferramenta <i>Manufacturer-defined</i> TEDS. ....	100
Figura B.6: Interface da ferramenta <i>Config Map Table</i> . ....	100
Figura B.7: Interface da ferramenta <i>Configure Weblab Modules Connections</i> . ....	101
Figura C.1: Modelo relacional da base de dados da aplicação desenvolvida . ....	105

# Lista de Tabelas

Tabela 2.1: Vantagens e desvantagens dos diferentes atores na experimentação remota.....	9
Tabela 2.2: Exemplos de laboratórios remotos e projetos. ....	16
Tabela 3.1: Família de normas IEEE 1451.x.....	23
Tabela 3.2: Tipos de TEDSs definidos pela norma IEEE 1451.0.....	26
Tabela 3.3: Formato genérico de um TEDS.....	27
Tabela 3.4: Exemplo de uma estrutura TLV. ....	28
Tabela 3.5: Estrutura do TEDS <i>Identification Header</i> . ....	29
Tabela 4.1: Formato genérico de um <i>Map Table</i> . ....	40
Tabela 5.1: Identificação do <i>User's Transducer Name</i> TEDS.....	53
Tabela 5.2: Formato final do <i>User's Transducer Name</i> TEDS.....	54
Tabela 5.3: Formato final do <i>TransducerChannel</i> TEDS. ....	60
Tabela 5.4: Formato final do <i>Manufacturer-defined</i> TEDS. ....	66
Tabela 5.5: Estrutura de dados UUID.....	69
Tabela 5.6: Conversão para binário da estrutura de dados UUID. ....	70
Tabela 5.7: Conversão para hexadecimal da estrutura de dados UUID. ....	70
Tabela 5.8: Formato final do <i>Meta</i> -TEDS. ....	72
Tabela 5.9: Definição do <i>Map Table</i> - TIM. ....	73
Tabela 5.10: Formato final do <i>Map Table</i> - Configuração 1. ....	73
Tabela 5.11: Formato final do <i>Map Table</i> - Configuração 2. ....	74
Tabela 5.12: Vantagens e desvantagens na utilização da aplicação desenvolvida face ao processo manual. ....	82
Tabela A.1: Estrutura do <i>Meta</i> -TEDS. ....	89
Tabela A.2: Estrutura do <i>TransducerChannel</i> TEDS. ....	90
Tabela A.3: Estrutura do <i>User's Transducer Name</i> TEDS. ....	94
Tabela A.4: Estrutura do <i>End User Application Specific</i> TEDS. ....	94





# Abreviaturas e acrónimos

<b>ADC</b>	<i>Analog-to-Digital Converter</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>CAN</b>	<i>Control Area Network</i>
<b>DAC</b>	<i>Digital-to-Analog Converter</i>
<b>FPGA</b>	<i>Field Programmable Gate Arrays</i>
<b>HDL</b>	<i>Hardware Description Language</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>IEEE</b>	<i>Institute of Electrical and Electronics Engineers</i>
<b>ISEP</b>	<i>Instituto Superior de Engenharia do Porto</i>
<b>JTAG</b>	<i>Joint Test Action Group</i>
<b>NCAP</b>	<i>Network Capable Application Processor</i>
<b>NIST</b>	<i>National Institute of Standards and Technology</i>
<b>RecTool</b>	<i>Reconfiguration Tool</i>
<b>SGBD</b>	<i>Sistema de Gestão de Base de Dados</i>
<b>SI</b>	<i>Sistema Internacional de Unidades</i>
<b>TC</b>	<i>TranducerChannel</i>
<b>TEDS</b>	<i>Transducer Electronic Data Sheet</i>
<b>TIC</b>	<i>Tecnologias da Informação e Comunicação</i>
<b>TII</b>	<i>Transducer Independent Interface</i>
<b>TIM</b>	<i>Transducer Interface Module</i>
<b>TLV</b>	<i>Type/Length/Value</i>
<b>URL</b>	<i>Uniform Resource Locator</i>
<b>USB</b>	<i>Universal Serial Bus</i>
<b>UUID</b>	<i>Universally Unique Identifier</i>
<b>XML</b>	<i>eXtensible Markup Language</i>



# Glossário

**Aplicação:** é uma ferramenta ou um conjunto de ferramentas concebidas para serem usadas pelo utilizador final.

**Experimentação remota:** permite a ligação de utilizadores a um conjunto de equipamentos reais, a qualquer hora e em qualquer lugar, através da *Internet/Intranet*.

**Ficheiro binário:** armazena os dados como sequências de *bits* e são processados como uma sequência de *bytes*. É necessário um editor binário para visualizar os seus valores.

**Ficheiro de texto:** armazena os dados como sequências de *bits* e são processados como uma sequência de caracteres. São representados no código ASCII, ou seja, podem ser lidos ou escritos por um qualquer processador de texto.

**Field Programmable Gate Arrays (FPGA):** é um dispositivo semiconductor configurável com diferentes módulos de laboratório.

**Groupware:** são aplicações que permitem a colaboração de vários utilizadores na realização de trabalhos de grupo.

**Infraestrutura laboratorial remota:** é responsável por implementar o sistema que controla e monitoriza os equipamentos e experiências laboratoriais através da *Internet/Intranet*.

**Laboratório híbrido:** é um laboratório que interliga instrumentos reais com instrumentos simulados por *software*.

**Laboratório remoto:** é um laboratório controlável através da *Internet/Intranet* para a realização de experiências. É implementado através de uma infraestrutura laboratorial remota.

**Laboratório remoto reconfigurável:** é um laboratório controlável através da *Internet/Intranet* para a realização de experiências cuja infraestrutura subjacente é reconfigurável com diferentes módulos (e.g. instrumentos de medição, robôs, motores, entre outros).

**Laboratório tradicional:** representa todos os laboratórios físicos que estão presentes em diversas instituições de ensino.

**Laboratório virtual:** é um laboratório simulado em computador através de aplicações de *software*.

**Módulos de laboratório:** é o conjunto de instrumentos e módulos embutidos na infraestrutura de um laboratório.

**Network Capable Application Processor (NCAP):** é o dispositivo entre o módulo do transdutor e a rede. Tem ainda a função de converter os dados recebidos pelo TIM.

**TransducerChannel (TC):** é o conjunto formado pelo transdutor e o condicionamento de sinal, possuindo um endereço de destino utilizado pelos comandos enviados do NCAP para o TIM.

**Transducer Electronic Data Sheets (TEDS):** são estruturas de dados que armazenam informação relacionada com os transdutores (e.g. identificação do fabricante, o tipo de transdutor, os canais do transdutor, número de série, entre outros).

**Transducer Interface Module (TIM):** é o módulo que tradicionalmente contém os TEDSs, o condicionamento de sinal, o conversor de dados e implementa ainda a interface lógica com os TCs.

# 1 Introdução

Atualmente é importante olhar a educação como um processo global que pode ser melhorado através da tecnologia. Inicialmente, as ferramentas educacionais, disponíveis para professores e alunos, apenas eram usadas nas salas de aula tradicionais. Neste momento, a facilidade de utilização da *Internet* tem promovido a adoção das tecnologias, principalmente nos cursos de Engenharia, através da disponibilização de laboratórios remotos [2]. Desta forma, a tecnologia desempenha um duplo papel na educação, pois facilita o processo de ensino e aprendizagem, bem como motiva os professores e alunos a interagirem e a realizarem experiências laboratoriais à distância.

Neste capítulo é feito o enquadramento deste trabalho e apresentados os motivos para a sua realização. De seguida são enumerados os objetivos e as suas principais contribuições. Conclui-se o capítulo com a descrição das considerações de escrita desta dissertação, bem como a apresentação de cada um dos seus capítulos e anexos.

## 1.1 Enquadramento e motivação

O processo de aprendizagem através da tecnologia tem contribuído para mudanças nos comportamentos sociais. Neste momento, a quantidade de informação disponível para ser consultada coloca pressão no sentido das pessoas se manterem atualizadas, de forma a prevenir um afastamento cultural e social do resto da sociedade. Nesta era digital, a informação circula livremente através da *Internet* cada vez mais diversificada e atualizada. Qualquer pessoa consegue consultar esta informação com o recurso a um computador ou a um dispositivo móvel, independentemente do local onde se encontra [3].

A evolução tecnológica tem também contribuído para o aparecimento de novas ferramentas de ensino e aprendizagem com o objetivo de melhorar a educação atual. Estas ferramentas têm a função de complementar o ensino tradicional, que é feito nas salas de aula, com um trabalho experimental realizado através de laboratórios remotos. Desta forma, o recurso à

tecnologia facilita uma maior interação de alunos e professores, permitindo que estes possam interagir, partilhar ideias, estratégias e realizar experiências remotas através da *Internet* [3]. Assim, os laboratórios remotos são um importante recurso para complementar os ambientes de ensino. No entanto, as infraestruturas dos laboratórios remotos atuais são desenvolvidas seguindo uma implementação específica, com várias arquiteturas de *hardware* e *software*, que usam diferentes linguagens de programação para controlar remotamente os instrumentos e módulos, resumidamente denominados por módulos de laboratório, que estão embutidos na infraestrutura de um laboratório. A inexistência de uma norma que regularize esta implementação tem, por isso, dificultado a utilização e divulgação destes laboratórios devido às seguintes limitações [4][5]:

- Não promove a colaboração entre as instituições de ensino, devido à dificuldade de reutilização e partilha dos módulos de laboratório de cada experiência;
- Elevado custo na implementação da infraestrutura de um laboratório remoto;
- Falta de conhecimentos técnicos, por parte das instituições de ensino, para desenvolver um laboratório remoto;
- Dificuldade na atualização de um laboratório remoto quando este é baseado num simples computador, devido à existência de vários *softwares* para realizar as diferentes experiências.

Para ultrapassar estas limitações foi desenvolvido no âmbito de um trabalho de doutoramento [1] um protótipo, cuja arquitetura é baseada na norma IEEE 1451.0, que permite a reconfiguração da infraestrutura do laboratório remoto recorrendo à utilização de um dispositivo lógico programável, nomeadamente de uma FPGA, para a integração dos vários módulos de laboratório. Desta forma, é possível embutir os diferentes módulos nos blocos de memória de uma FPGA, tornando desnecessária a utilização dos módulos de laboratório tradicionais [6]. No entanto, a preparação de cada módulo de laboratório obriga à criação de, pelo menos, um TEDSs, bem como a definição de outros ficheiros que interligam estes módulos com a infraestrutura do laboratório. Os TEDSs são estruturas de dados que armazenam informação relacionada com os transdutores, cuja estrutura é definida de acordo com a norma IEEE 1451.0 [7]. Contudo, a complexidade subjacente para a definição desses ficheiros incentivou à implementação de uma aplicação *Web* para facilitar a leitura e escrita destes ficheiros, bem como dos ficheiros que permitem embutir os diferentes módulos de laboratório na infraestrutura do laboratório.

## 1.2 Objetivos e principais contribuições

A reconfiguração da infraestrutura do laboratório remoto é feita através de uma ferramenta de configuração. Esta ferramenta tem o objetivo de reconfigurar a infraestrutura do laboratório remoto já desenvolvido através da submissão de ficheiros de *Configuração*, que interligam os vários módulos de laboratório com a infraestrutura, e de *Projeto*, que descrevem

cada módulo de laboratório, bem como os TEDSs utilizados para a compilação de um projeto de um laboratório remoto [5]. Ambos os ficheiros têm uma estrutura variável e pressupõem o conhecimento do utilizador para a sua definição. Para além disso, a configuração destes ficheiros exige ao utilizador a realização de vários cálculos e conversões de acordo com a norma IEEE 1451.0.

Neste contexto, a principal limitação da ferramenta de reconfiguração de laboratórios remotos reside na dificuldade de escrever e ler os ficheiros de *Configuração* e de *Projeto*. Assim, esta dissertação propõe solucionar este problema, com o desenvolvimento de uma aplicação *Web* para facilitar a leitura e escrita destes ficheiros. Desta forma, é possível referir, no âmbito desta dissertação, os seguintes contributos:

- Estudo da norma IEEE 1451.0, com particular enfoque para a sua arquitetura e para a estrutura dos diferentes TEDSs;
- Estudo do protótipo do laboratório remoto reconfigurável;
- Desenvolvimento de uma aplicação *Web* que permita a leitura e escrita de TEDSs, criados segundo a norma IEEE 1451.0, bem como dos ficheiros que interligam os módulos de laboratório com a infraestrutura do laboratório.

### **1.3 Considerações de escrita**

Durante a escrita deste documento foram considerados os seguintes aspetos:

- A primeira referência de cada abreviatura ou acrónimo é sucedida do seu significado entre parêntesis;
- As citações apresentadas no documento surgem entre aspas;
- O documento foi escrito ao abrigo do novo acordo ortográfico;
- Os endereços *Web* apresentados nas notas de rodapé pretendem complementar a informação referida na própria página.

### **1.4 Organização do documento**

O documento é composto por sete capítulos e quatro anexos, cujo conteúdo será brevemente explicado. Salienta-se ainda a existência de um DVD com a aplicação.

No primeiro capítulo é realizado o enquadramento desta dissertação, sendo identificados os objetivos, bem como as principais contribuições.

No segundo capítulo é dada uma visão global sobre a experimentação remota, descrevendo-se os seus principais objetivos, bem com os impactos da sua utilização para alunos, professores e instituições de ensino.

No terceiro capítulo é descrito resumidamente as normas IEEE 1451.x, com particular enfoque para a norma IEEE 1451.0 e a arquitetura associada. Descreve-se a arquitetura de um protótipo de um laboratório remoto reconfigurável e é ainda apresentada uma ferramenta que permite a sua reconfiguração.

No quarto capítulo é realizada a contextualização da aplicação desenvolvida no âmbito desta dissertação. Descreve-se a sua arquitetura, as principais funcionalidades, e apresentam-se as interfaces das diferentes ferramentas disponibilizadas na aplicação. Por fim, descrevem-se os aspetos desenvolvimento necessários para a sua criação.

No quinto capítulo é feita a verificação experimental, apresentando-se um cenário prático para realizar uma análise qualitativa da aplicação em relação à leitura e escrita dos ficheiros de forma manual.

No sexto capítulo é resumido o trabalho realizado neste projeto, bem como as suas limitações e propostas para futuras melhorias.

Por fim, são enumeradas as referências utilizadas neste trabalho, que precedem o seguinte capítulo de anexos:

- Anexo A: descrição da estrutura dos vários TEDSs que a aplicação desenvolvida permite ler e escrever;
- Anexo B: apresentação das várias interfaces *Web* criadas para cada uma das ferramentas disponíveis na aplicação;
- Anexo C: apresentação do modelo relacional da base de dados utilizada;
- Anexo D: apresentação dos vários ficheiros XML (*eXtensible Markup Language*) utilizados para validar os diferentes campos de cada uma das ferramentas disponíveis na aplicação.

Importa ainda referir da existência de um DVD com os recursos da aplicação desenvolvida e com a descrição das configurações necessárias para a sua instalação e utilização.



## 2 Experimentação remota no ensino

Nos últimos anos, as novas Tecnologias da Informação e Comunicação (TIC) têm servido de apoio à educação. Aos alunos é possibilitado o acesso a recursos pedagógicos através de um computador com ligação à *Internet*. Se antes, estes recursos se limitavam a fornecer acesso a um conjunto de informação documental (e.g. jornais, livros, artigos, entre outros), hoje são várias as ferramentas que possibilitam o acesso a outras tecnologias de apoio ao ensino, tais como ambientes de ensino virtual e mais recentemente os laboratórios remotos que possibilitam a realização de trabalhos experimentais à distância [8].

Neste capítulo é explicada a importância do ensino à distância. É definido o conceito de experimentação remota, e as vantagens e desvantagens na sua utilização no ensino, quer para alunos, professores e para as próprias instituições de ensino. É explicada a arquitetura de um laboratório remoto tradicional e a organização da infraestrutura subjacente denominada por infraestrutura laboratorial remota. Para auxiliar esta explicação, são apresentados alguns projetos envolvendo a utilização de laboratórios remotos. Por fim, são enumerados alguns dos problemas atuais na utilização e construção de infraestruturas laboratoriais remotas.

### 2.1 O ensino à distância

A partir do momento em que a transmissão de conhecimento a grupos de alunos foi atribuída a um professor, iniciaram-se diversas pesquisas e teorias relacionadas com a aprendizagem humana. Em todos os estudos realizados verificou-se que a aprendizagem humana dependia de duas componentes fundamentais: a teórica e a prática [9].

A componente teórica nasce com a transmissão do conhecimento. É através desta que o professor transmite os seus conhecimentos aos alunos utilizando referências teóricas, conceitos e definições [9][10]. Nesta componente é fundamental, principalmente nas áreas das ciências e engenharias, assegurar que cada teoria lecionada se relacione com o mundo real.

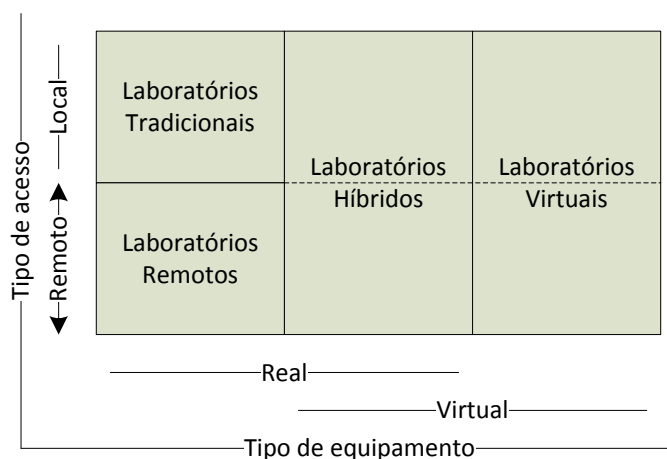
A componente prática é responsável por desenvolver as capacidades técnicas adquiridas durante o ensino da componente teórica, tais como: a compreensão de um problema, a sua simplificação e respetivo planeamento de resolução. A componente prática está dividida em duas partes: i) exercícios práticos e ii) trabalho experimental. Seja nas aulas ou em casa, os exercícios práticos são importantes para inculcar hábitos de trabalho e autonomia no estudo dos alunos. No entanto, estes exercícios são realizados com base em exemplos hipotéticos, o que desmotiva os alunos afastando-os da realidade. Para promover a motivação e o interesse na componente prática existe o trabalho experimental. Este possibilita uma maior aproximação com a realidade [8][10]. Para além disso, durante a sua condução e até à resolução do problema, é ainda exigido ao aluno a capacidade de formulação e verificação de hipóteses, observação e análise dos dados, elaboração de conclusões, bem como a familiarização com o uso de equipamentos e técnicas laboratoriais. Desta forma, é importante disponibilizar aos alunos ambientes laboratoriais para que estes possam experimentar e praticar o que aprenderam durante as aulas teóricas.

O melhoramento das tecnologias e dos serviços suportados pela *Internet* permitiu a criação de vários ambientes laboratoriais para que os alunos possam realizar as suas experiências. Tal como é ilustrado na figura 2.1, atualmente os laboratórios são classificados de acordo pelo seu tipo de acesso (local ou remoto) e equipamento (real ou virtual), cada um com características específicas [6][8][11]:

- Laboratório tradicional: representa os laboratórios comuns criados antes da chegada da *Internet*. Nestes laboratórios os alunos têm acesso a diversos instrumentos utilizados para a realização de experiências. A utilização destes laboratórios está condicionada por limitações de espaço, disponibilidade de um profissional responsável para acompanhar as experiências, quantidade de instrumentos, horários de acesso e localização geográfica, entre outras. Além disso, outro motivo que dificulta o trabalho experimental nestes laboratórios é o custo avultado na sua implementação e manutenção em cada instituição de ensino [3].
- Laboratório virtual: cria um local de trabalho em que todos os instrumentos são simulados em computador através de programas de *software*. As interfaces desenvolvidas contêm ambientes gráficos interativos para simular a realização de experiências, transmitindo aos alunos a sensação de controlarem equipamento real. A ligação com o laboratório virtual é local ou remota. Para simularem a realização das experiências os alunos descarregam para o computador o *software* específico ou acedem diretamente a um laboratório virtual através da *Internet* [8][11]. Apesar destes laboratórios serem importantes no ensino, não devem ser utilizados como uma alternativa aos laboratórios tradicionais, mas sim como um complemento.
- Laboratório híbrido: permite o acesso local ou remoto a equipamentos reais ou virtuais. No acesso remoto, o aluno pode usar um dispositivo (e.g. computador, *tablet*, entre outros) para aceder a uma experiência através da *Internet*, onde, durante a interação com os instrumentos, algumas partes da experiência podem ser reais e outras simuladas por *software*. No acesso local, o laboratório híbrido engloba alguns instrumentos reais, capazes de serem controlados localmente, como num

laboratório tradicional, mas que contém algumas partes simuladas por computador. Apesar destes ambientes serem incomuns, devem ser considerados para os momentos em que os instrumentos são demasiado caros ou estão indisponíveis, e para as situações em que o resultado da experiência é impossível de visualizar (e.g. visualização das linhas de um campo magnético) [3].

- Laboratório remoto: é implementado através de uma infraestrutura laboratorial remota com o objetivo de disponibilizar um laboratório controlável através da *Internet/Intranet* para a realização de experiências. Um laboratório remoto permite o acesso dos alunos a um conjunto de equipamentos reais, a qualquer hora e em qualquer lugar. Desta forma, tal como num laboratório tradicional, o aluno tem ao seu dispor um ambiente capaz de interpretar os seus dados e de realizar a experiência pretendida, com a possibilidade desta ser observada e monitorizada durante o seu funcionamento [8][11] .



**Figura 2.1: Tipos de ambientes laboratoriais.**

(Imagem retirada de R. Costa et al., "Reconfigurable weblabs based on the IEEE 1451 Std.", 2010).

A escolha por um destes ambientes laboratoriais para o apoio ao ensino, depende de vários contextos educativos, de acordo as instituições de ensino, os requisitos dos cursos e o tipo de professores e alunos que vão utilizar o laboratório. Como os laboratórios remotos são um tema importante para o trabalho descrito nesta dissertação, a sua importância no apoio ao ensino será resumidamente abordada na secção seguinte.

## 2.2 Os laboratórios remotos no apoio ao ensino

Um laboratório remoto permite o acesso de diversos atores (alunos, professores) a um conjunto de equipamentos reais a qualquer hora e a partir de qualquer lugar com ligação à *Internet/Intranet*. Na utilização destes equipamentos é importante garantir a entreaajuda e a interação entre alunos e professores. Os laboratórios remotos são assim um bom complemento na aprendizagem dos alunos e no trabalho dos professores, garantindo a flexibilidade em termos de localização, tempo e ritmo de trabalho [3].

### 2.2.1 Atores envolvidos

Os principais atores de um laboratório remoto são: i) os alunos, ii) os professores e iii) as instituições de ensino, cada um com características específicas:

- Instituições de ensino: com a disponibilização de laboratórios remotos os equipamentos envolvidos podem ser mais utilizados do que num laboratório tradicional, dado que não existem restrições de horários para a realização de experiências remotas. Desta forma, o investimento inicial da instituição na construção do laboratório remoto pode traduzir-se numa boa ferramenta de trabalho para os alunos e professores [3]. Apesar dos laboratórios remotos não serem distribuídos, estes podem ser partilhados entre diversas instituições de ensino. Cada instituição é responsável por construir e gerir o seu próprio laboratório, disponibilizando assim as experiências nas áreas de ensino da sua competência. Desta cooperação, alunos de diferentes instituições podem beneficiar da partilha de conhecimento e dos recursos disponibilizados, melhorando a sua aprendizagem nas diversas áreas [3][8].
- Professores: a facilidade de acesso aos laboratórios remotos permite melhorar a qualidade das suas matérias práticas e dos seus exercícios, colocando ao dispor dos seus alunos recursos que, por vezes, não estão disponíveis devido à falta de meios. Esta medida promove o envolvimento dos alunos num ambiente real, tornando-os mais ativos e relacionados com o meio, não sendo apenas meros observadores durante a experiência [8].
- Alunos: a flexibilidade de horários dos laboratórios remotos permite uma melhor organização do seu tempo e das suas tarefas, de forma a adequarem o seu ritmo de aprendizagem e de trabalho. Com a sua utilização, os alunos podem ainda aperfeiçoar a sua habilidade técnica, sendo uma boa forma para se prepararem para as avaliações de componente prática [11]. Como é exigido um trabalho mais autónomo aos alunos que realizam experiências remotas, surgem com mais frequência dúvidas e dificuldades na condução das experiências. Esta desvantagem, num laboratório tradicional não se faz notar, pois essas dúvidas são rapidamente esclarecidas recorrendo a um professor ou a um técnico responsável pelo laboratório. Para reduzir estes problemas, as experiências disponibilizadas deverão ser bem suportadas por documentação de apoio para quem as vai realizar [8][11].

### 2.2.2 Vantagens vs desvantagens

Para as instituições de ensino um benefício importante para a utilização de um laboratório remoto é a facilidade de disponibilizar experiências sempre atuais, auxiliando o trabalho dos professores no ensino [11]. No entanto, apesar de existir uma maior flexibilidade na disponibilização e atualização dessas experiências, a instalação e a manutenção de um laboratório remoto exige, pelo menos, um técnico financiado o que poderá pressupor custos associados.

Para além de alunos e professores, os laboratórios remotos estão também acessíveis a um qualquer utilizador. A estes é possibilitado o acesso remoto aos laboratórios reduzindo os custos de deslocamento associados. Outra vantagem é a possibilidade de um utilizador com deficiência física, nomeadamente motora, aceder remotamente a experiências sem a necessidade de se deslocar. Através de aplicações informáticas, o laboratório remoto pode ainda ser adaptável a outro tipo de deficiências mais impeditivas de uma normal utilização, como é o caso da deficiência visual, auditiva ou mesmo mental.

O acesso a laboratórios remotos através da *Internet* é por vezes uma desvantagem. Se por um lado o laboratório fica disponível para qualquer utilizador, por outro o controlo pela *Internet* aumenta a probabilidade de ocorrerem falhas durante a sua utilização. Situações em que a *Internet* se apresenta instável, em que o sistema está indisponível ou não responde aos pedidos dos utilizadores, ou simplesmente quando as experiências deixam de funcionar, são algumas das desvantagens na utilização dos laboratórios remotos. Por utilizar uma ligação remota, pode não ser possível reiniciar novamente a experiência, sendo necessário que o técnico do laboratório a reinicie localmente. Felizmente, os desenvolvimentos tecnológicos têm permitido que os sistemas sejam cada vez mais sofisticados e fiáveis, reduzindo dessa forma os problemas de instabilidade e de resposta de ligação na utilização dos laboratórios remotos [11].

Outra questão a considerar reside na mentalidade dos atores de um laboratório remoto. Como o ensino tradicional está muito enraizado e os professores ainda estão muito dependentes dos laboratórios tradicionais, estes muitas vezes ignoraram a possibilidade de utilizar os laboratórios remotos como um método auxiliar de ensino [11][12].

Com base nas considerações apresentadas nos parágrafos anteriores, a tabela 2.1 resume as vantagens e desvantagens na utilização de um laboratório remoto para os diferentes atores envolvidos no processo do ensino e aprendizagem.

**Tabela 2.1: Vantagens e desvantagens dos diferentes atores na experimentação remota.**

	<b>Alunos</b>	<b>Professores</b>	<b>Instituições de ensino</b>
Vantagens	Acesso a qualquer hora.	Acesso a qualquer hora.	Facilidade de gestão.
	Facilidade de acesso independente do lugar.	Facilidade de acesso independente do lugar.	Maior utilização dos equipamentos.
	Aprendizagem ao seu ritmo e autónoma.	Organização dos seus tempos de trabalho.	Redução dos custos.
	Acesso a documentação detalhada de cada experimentação.	Possibilidade de incluírem experiências práticas nas suas aulas.	Partilha de conhecimento e partilha de equipamentos de diferentes áreas.
	Envolvimento num ambiente real e motivador.	Possibilidade de criarem exercícios para grupos de alunos.	Facilidade de atualização das experiências remotas.
	Facilidade de acesso para alunos portadores de deficiência física.	Facilidade para os professores com deficiência física lecionarem cursos à distância.	Cooperação com outras instituições.
	Melhoramento da sua habilidade técnica.		

Desvantagens	Necessidade de <i>Internet</i> estável.	Necessidade de <i>Internet</i> estável.	Custos na implementação de um sistema fiável.
	Dificuldade para reiniciar uma experiência que não responda aos pedidos.	Dificuldade para reiniciar uma experiência que não responda aos pedidos. Ensino tradicional ainda muito enraizado.	Financiamento de um técnico especializado.

### 2.2.3 Ferramentas de comunicação adotadas

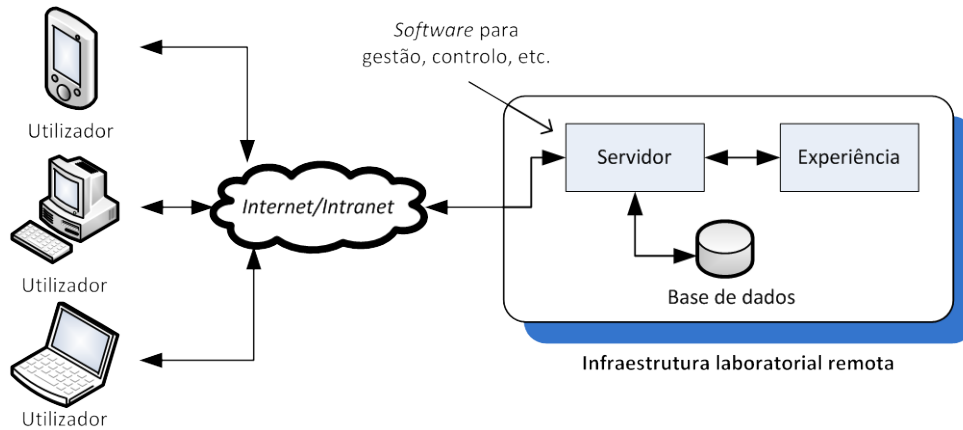
Sempre que possível, um laboratório remoto deverá garantir recursos semelhantes aos disponibilizados por um laboratório tradicional. Com este objetivo, devem ser disponibilizadas várias ferramentas de comunicação, que promovem o trabalho em grupo sobre uma dada experiência. Para os atores envolvidos no processo de ensino e aprendizagem, estas ferramentas, designadas por *groupware*, permitem a troca de ideias, sugestões e opiniões entre si, independentemente da sua localização ou tipo de comunicação pretendida, que poderá ser de dois tipos [8]:

- Comunicação síncrona: é um tipo de comunicação realizada em tempo real entre dois ou mais atores, podendo ser implementada através de videoconferência ou por comunicação escrita (e.g. *chat*). Permite que os atores interajam em tempo real durante a realização de uma experiência. Se por um lado, esta forma de comunicação permite aos alunos um modo mais eficiente na troca de ideias entre si ou com os professores a qualquer momento, por outro lado, obriga a uma disponibilidade simultânea de ambos.
- Comunicação assíncrona: é um tipo de comunicação diferida no tempo, e que pode ser implementada através da troca de ficheiros ou de mensagens de correio eletrónico entre os atores. Não aproxima o laboratório remoto do laboratório tradicional, na medida que não impõe uma resposta imediata à informação trocada, nem obriga a uma atenção simultânea de ambos os atores. Em comparação com o outro tipo de comunicação garante uma maior flexibilidade temporal aos diferentes atores, uma vez que não necessita que estejam simultaneamente ligados ao laboratório remoto.

## 2.3 Arquitetura de um laboratório remoto

A arquitetura de um laboratório remoto é suportado por uma infraestrutura laboratorial remota para disponibilizar o controlo e monitorização de um conjunto de equipamentos, genericamente denominados por módulos de laboratório, e experiências laboratoriais. No momento da construção de um laboratório remoto é fundamental definir uma arquitetura que possibilite o acesso remoto a experiências e aos equipamentos utilizados. A arquitetura que melhor se adequa a este princípio é definida por um modelo do tipo cliente-servidor, ilustrado na figura 2.2. Neste modelo o servidor é a localização central onde os utilizadores acedem. Este centraliza os serviços da rede oferecidos aos utilizadores, tais como aplicações

ou filas de impressão, numa única localização. Apesar de exigir um maior esforço na sua gestão e implementação, o modelo cliente-servidor tem vantagens, nomeadamente no que se refere à flexibilidade no acesso dos utilizadores ao servidor da infraestrutura [8][13][14].



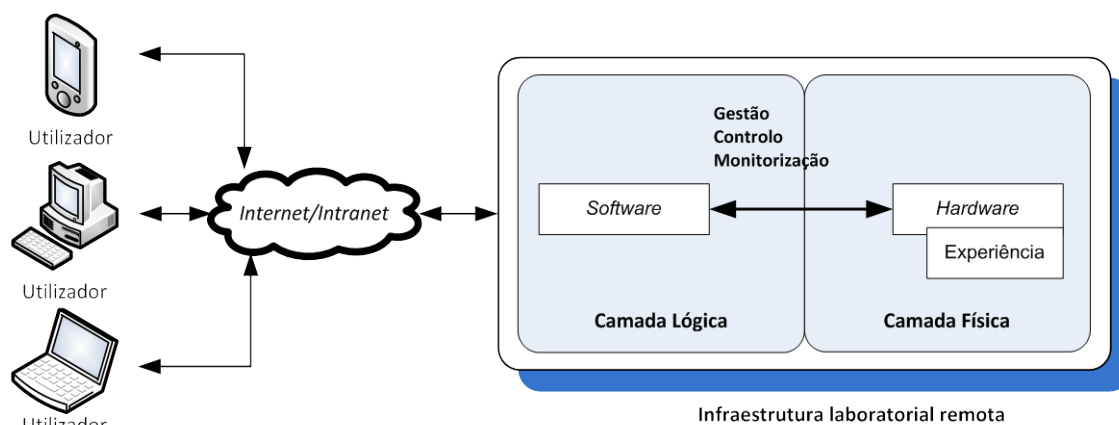
**Figura 2.2: Arquitetura cliente-servidor utilizada no acesso a uma infraestrutura laboratorial remota.**

A arquitetura de um laboratório remoto é composta três partes: i) atores, que representam todos os utilizadores do laboratório remoto, ii) servidor, que tem as funções de controlar e monitorizar a infraestrutura laboratorial remota e, iii) um servidor que guarda todos os dados necessários através de um SGBD (Sistema de Gestão de Base de Dados). O servidor é a parte mais importante desta arquitetura. É importante garantir a qualidade da tecnologia implementada no servidor para não prejudicar todo o sistema [13].

Cada um dos atores tem a responsabilidade de iniciar a comunicação com o servidor. Ao iniciar os pedidos com o servidor, cada ator envia comandos para controlar os equipamentos remotos, e dados estruturados para efetuar configurações específicas na infraestrutura laboratorial remota. Por outro lado, o servidor tem a responsabilidade de responder às ações dos atores com os dados solicitados, nomeadamente com o resultado das experiências [8][14]. O servidor tem ainda a missão de estabelecer o canal de comunicação entre os atores e as experiências remotas disponibilizadas pela infraestrutura laboratorial remota. O servidor fornece também o material de suporte para a realização de determinada experiência, incluindo as respetivas interfaces que permitam o controlo das suas características [14]. Assim, a infraestrutura de um laboratório remoto deve permitir [12]:

- O controlo e monitorização de experiências;
- O fornecimento de informação detalhada sobre a realização de experiências;
- A transmissão em direto da realização de cada experiência, sempre que possível;
- A comunicação entre os utilizadores;
- A recuperação de falhas;
- A segurança dos equipamentos disponíveis, garantindo adequadamente quais os atores que acedem a cada um deles;
- A boa qualidade da largura de banda do laboratório, para que a ligação remota não seja prejudicada.

Como ilustrado na figura 2.3, a organização da infraestrutura laboratorial remota está dividida em duas camadas: i) camada de *software* (camada lógica), que é a responsável por monitorizar e gerir todos os recursos necessários, e ii) camada de *hardware* (camada física), que refere a todos os equipamentos reais utilizados na construção do laboratório remoto (servidor e experiências) [8].



**Figura 2.3: Camada lógica e camada física de uma infraestrutura laboratorial remota.**

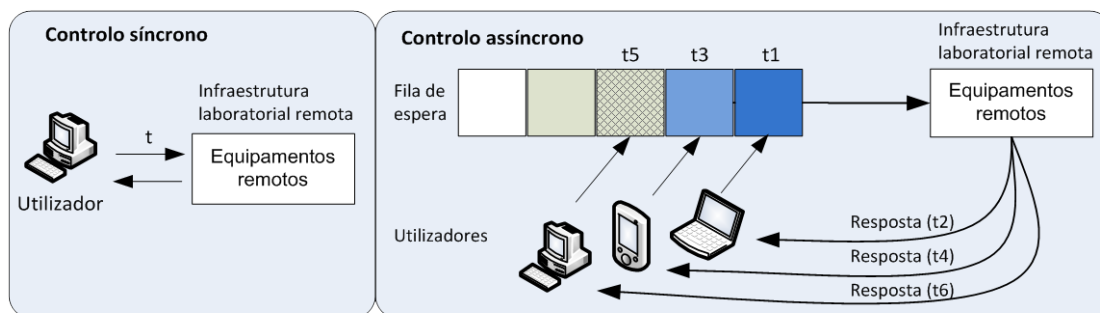
### 2.3.1 Camada lógica

A camada lógica é responsável pelo controlo, gestão e monitorização da infraestrutura, conforme representado na figura 2.4 [8]. O desenvolvimento da camada lógica deverá garantir uma comunicação síncrona ou assíncrona com os equipamentos remotos, cada uma com características específicas:

- Controlo síncrono: permite aos utilizadores uma ligação em tempo real às experiências disponibilizadas pelo laboratório remoto. Esta comunicação síncrona só será possível se o laboratório remoto estiver livre, ou seja, as mesmas experiências não podem estar alocadas por outros utilizadores. Para garantir que o acesso é realizado no momento previsto, deverá ser efetuada uma marcação prévia do horário pretendido;
- Controlo assíncrono: permite uma gestão mais eficaz dos equipamentos remotos, possibilitando a qualquer instante o acesso às experiências sem a necessidade de efetuar previamente as reservas. Para a implementação deste tipo de contacto é essencial que todos os pedidos efetuados sejam geridos por uma fila de espera, até ao momento em que o laboratório fica livre e os dados sejam transferidos para serem usados.

Os objetivos da camada lógica garantem ainda a flexibilidade do acesso dos utilizadores remotos, utilizando um modelo *Thin Client* ou *Thick Client*.





**Figura 2.4: Controlo síncrono e assíncrono de uma infraestrutura laboratorial remota.**

(Imagem retirada de R. Costa, "Infra-estrutura laboratorial para a experimentação remota", 2003).

No modelo *Thin Client* o utilizador acede ao laboratório remoto através de interfaces executadas num simples *Web browser* (e.g. Internet Explorer, Google Chrome, entre outros). Para auxiliar esta ligação será necessário a instalação de extensões (*plug-ins*) para adicionar novas funcionalidades específicas ao *Web browser* escolhido. A comunicação dos atores com os laboratórios remotos é efetuada pelas interfaces que são descarregadas no momento da ligação com a infraestrutura laboratorial remota, através de um endereço *Web* específico (*Uniform Resource Locator*, URL). Este modelo permite ainda que as atualizações sejam feitas apenas pelo servidor *Web*, garantindo interfaces sempre atuais [8][13].

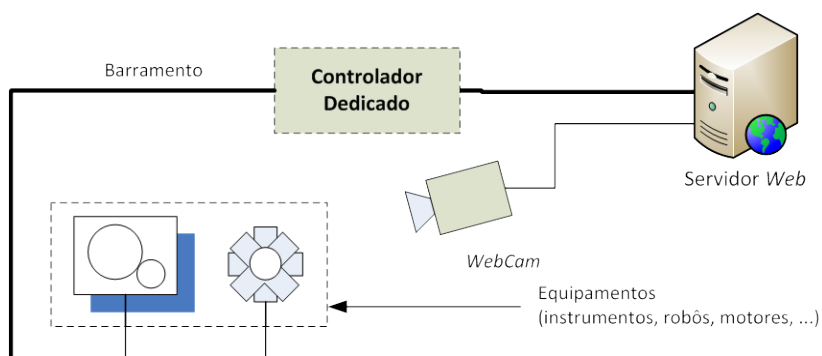
No modelo *Thick Client* as interfaces que controlam o laboratório remoto são instaladas nos computadores dos utilizadores. As aplicações podem ser desenvolvidas em várias linguagens (C, C++, Java, .NET, Delphi, Python, entre outras) e devem suportar diferentes sistemas operativos, o que pode significar algumas restrições de utilização. Nesta solução, o processo de atualização das interfaces torna-se mais complexo e demorado pois obriga a que as alterações sejam feitas em todos os computadores utilizados no acesso ao laboratório remoto. Outro inconveniente reside na intrusão das aplicações a instalar no computador. Ao estarem instaladas na máquina do utilizador, estas interfaces têm acessos privilegiados ao disco rígido, com permissões para ler e escrever qualquer tipo de ficheiro, ou até mesmo abrir novas ligações na *Internet*. Por outro lado, estas aplicações locais podem ter uma melhor qualidade gráfica do que as interfaces *Thin Client*, aproveitando os recursos disponíveis na máquina do próprio utilizador (e.g. interfaces 3D) [13].

### 2.3.2 Camada física

A construção da camada física de uma infraestrutura laboratorial remota depende das experiências disponibilizadas. Os equipamentos adotados deverão controlar localmente as experiências, com base nas instruções recebidas pelos utilizadores. Para além do servidor *Web*, que controla a ligação entre os utilizadores e as experiências, é importante adotar um equipamento que permita o retorno visual e sonoro das experiências conduzidas pelos utilizadores (e.g. *Webcam*) [8][15].

Por se tratar de um laboratório remoto, é fundamental que todos os equipamentos trabalhem autonomamente com base nas instruções recebidas dos utilizadores. Desta forma, é

necessário que toda a infraestrutura esteja interligada em rede, como é representado na figura 2.5.



**Figura 2.5: Interligação de equipamentos numa infraestrutura laboratorial remota.**  
(Imagem retirada de R. Costa, "Infra-estrutura laboratorial para a experimentação remota", 2003)

A ligação desta infraestrutura à *Internet/Intranet* é feita por um servidor *Web*, permitindo seleccionar os equipamentos necessários à realização de uma determinada experiência utilizando dois tipos de barramentos [8]:

- Barramento de Controlo (*Control Bus*): utilizado para interligar sensores e atuadores usando comunicações físicas muito simples e específicas para a troca de pequenas quantidades de dados a elevadas velocidades (e.g. CAN - *Control Area Network*<sup>1</sup>);
- Barramentos de instrumentação (*Instrumentation Bus*): utilizado para interligar vários módulos de laboratório controlados por um computador para a troca de dados mais complexos a velocidades elevadas (e.g. PXI - *PCI eXtensions for Instrumentation*<sup>2</sup> ou LXI - *LAN eXtensions for Instrumentation*<sup>3</sup>).

A maioria dos módulos de laboratório ligados a uma infraestrutura deste género tem a opção de controlo remoto através de telemanipuladores. Estes são equipamentos robotizados controlados pelos utilizadores à distância, funcionando como uma extensão do próprio braço (e.g. garra mecânica). A ligação destes módulos de laboratório é feita por um conector definido por *General Purpose Interface Bus* (GPIB<sup>4</sup>) ou ligados através da sua porta de rede (conector RJ45). O telemanipulador é, por vezes, o componente mais complicado de interligar num laboratório remoto. Este deve ser projetado para cada tipo de laboratório, usando vários atuadores, para transformarem o sinal de entrada em movimento (e.g. motores elétricos, relés, entre outros) [13].

A partir do momento em que o laboratório remoto é criado, este fica disponível a qualquer utilizador. A sua disponibilidade é influenciada pelo número de utilizadores conectados ao servidor, não sendo possível garantir uma comunicação síncrona com as experiências

<sup>1</sup> <http://www.iso.org/iso/search.htm?qt=Controller+Area+Network&published=true>

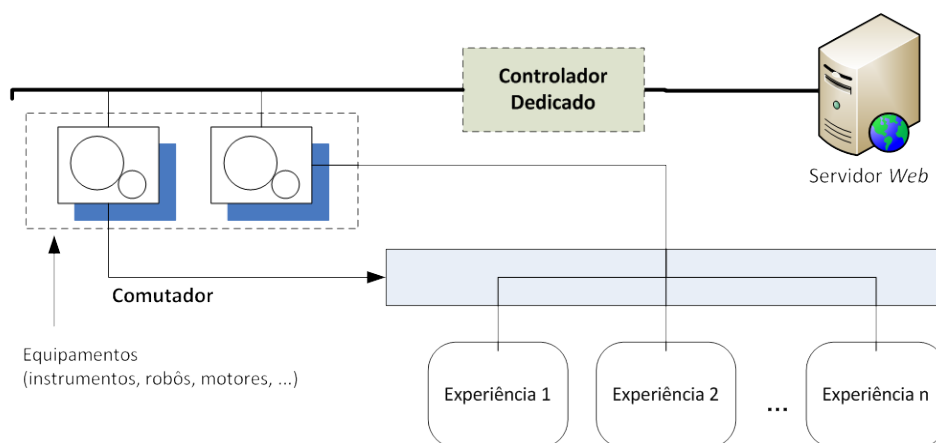
<sup>2</sup> <http://www.pxisa.org>

<sup>3</sup> <http://www.lxistandard.org/>

<sup>4</sup> <http://www.ni.com/white-paper/3419/pt/>

remotas, sem uma marcação prévia. Uma das formas de resolver este problema seria replicar as experiências existentes e os respectivos equipamentos, interligando-os a um servidor, ou então, replicar a própria infraestrutura laboratorial remota. Apesar das vantagens na fiabilidade e na disponibilidade da infraestrutura, os custos envolvidos desta replicação tornariam inviável a implementação desta solução. Assim, caso a disponibilidade contínua da infraestrutura não seja uma condição indispensável, poder-se-á optar por uma solução assíncrona, garantindo que todas as experiências sejam executadas através de uma fila de pedidos submetidos pelos utilizadores.

O funcionamento do laboratório remoto deve ser versátil. Para isso, a implementação da infraestrutura deve permitir a troca automática de experiências, sem a necessidade da intervenção dos utilizadores ou dos técnicos do laboratório. A alteração dos equipamentos entre as várias experiências deve ser realizada através de uma comutação automática, controlada por um equipamento específico, como é ilustrado na figura 2.6 [8].



**Figura 2.6: Sistema de comutação entre várias experiências numa infraestrutura laboratorial remota.**  
(Imagem retirada de R. Costa, "Infra-estrutura laboratorial para a experimentação remota", 2003)

A construção desta infraestrutura depara-se com um inconveniente, pois impossibilita que várias experiências sejam executadas em simultâneo. Se não houver esse interesse, esta infraestrutura poderá ser uma boa solução para reduzir os custos da sua implementação uma vez que todos os equipamentos podem ser reutilizados em várias experiências.

Todas estas considerações deverão ser analisadas para beneficiar alunos, professores e as instituições de ensino que disponibilizam as experiências. Neste contexto existem vários projetos e consórcios que fornecem uma visão genérica dos laboratórios remotos existentes.

## 2.4 Exemplos de laboratórios remotos

Atualmente vários projetos e consórcios internacionais procuram fornecer e definir várias arquiteturas de laboratórios remotos. Alguns desses projetos nasceram com o objetivo de implementar uma rede partilhada de laboratórios remotos entre diferentes instituições de ensino, permitindo o acesso a qualquer utilizador (e.g. Labshare, RexLab e Virtual Labs).

Foram ainda criados outros projetos com o propósito de definir infraestruturas que permitem a partilha de equipamento laboratorial e de material educacional entre vários laboratórios (e.g. MIT iLabs e VISIR). Alguns consórcios foram também criados com o objetivo de agrupar vários laboratórios remotos, funcionando como um repositório central de informação (e.g. Lab2go e LiLa). Existe ainda outros consórcios com a finalidade de definir uma norma para a unificação dos vários laboratórios remotos, nomeadamente sobre a sua implementação e acesso (e.g. GOLC e NSLOL WG). Na tabela 2.2 são descritos o nome, o ano de criação e o objetivo de cada projeto mencionado.

**Tabela 2.2: Exemplos de laboratórios remotos e projetos.**

Nome	Ano	Objetivo
<b>GOLC<sup>5</sup></b> <i>Global Online Laboratory Consortium</i>	2011	Tem como missão a divulgação de ambientes experimentais, procurando um mecanismo eficiente para os partilhar, com o objetivo de criar uma rede global de experiências. Procura incentivar e apoiar a criação de novos laboratórios via <i>Web</i> , assim como, dos respetivos materiais curriculares. Por fim, tem, ainda, como objetivo a criação de uma norma para a unificação da arquitetura dos laboratórios remotos, mais concretamente, sobre a sua implementação e acesso [6].
<b>Lab2go<sup>6</sup></b>	2010	Surgiu com o propósito de criar uma estrutura para reunir e descrever vários laboratórios remotos ou virtuais, localizados em todo o mundo, funcionando como um repositório central de informação. O portal Lab2go guarda os dados sobre os projetos de investigação e dos seus investigadores, as organizações e o estado da arte das tecnologias de cada competência. Esta plataforma <i>Web</i> pretende aproximar alunos, professores e investigadores sobre a partilha de conhecimento e de experiências em laboratórios remotos e virtuais [5].
<b>Labshare<sup>7</sup></b> <i>National Support for Laboratory Resource Sharing</i>	2009	Financiado pelo governo Australiano, foi um projeto que nasceu com o intuito de criar uma rede nacional partilhada de laboratórios remotos. Tem o objetivo de melhorar a qualidade dos laboratórios remotos, disponibilizando-os depois para os estudantes de qualquer lugar do mundo [5].
<b>LiLa<sup>8</sup></b> <i>Library of Labs</i>	2009	O projeto LiLa foi uma iniciativa de oito faculdades e de três empresas, por toda a europa, para a partilha de acessos a laboratórios remotos e virtuais. Este projeto deu lugar à criação de um portal que, para além de permitir o acesso a estes laboratórios, possui um sistema para agendar a sua utilização. Além deste portal, o projeto LiLa criou uma estrutura organizacional para a partilha de experiências entre diferentes instituições e para o acesso a configurações experimentais [5].
<b>MIT iLabs<sup>9</sup></b>	2006	Localizado no <i>Massachusetts Institute of Technology</i> (MIT), o projeto iLabs disponibiliza vários laboratórios remotos sobre ciência e engenharia. Tem como visão a partilha de equipamentos caros e de material educacional em todo o mundo. A sua arquitetura de partilha ( <i>iLabs Shared Architecture, ISA</i> ) é uma infraestrutura que pretende minimizar o esforço de desenvolvimento e gestão dos laboratórios

<sup>5</sup> Projeto disponível em <http://www.online-lab.org/>

<sup>6</sup> Projeto disponível em <http://www.lab2go.net>

<sup>7</sup> Projeto disponível em <http://www.labshare.edu.au/project>

<sup>8</sup> Projeto disponível em <https://www.library-of-labs.org/startPage/startPage.action>

<sup>9</sup> Projeto disponível em <http://web.mit.edu/edtech/casestudies/ilabs.html>

		remotos, fornecendo assim um conjunto de serviços e ferramentas para a sua construção [16].
<b>NSLOL WG<sup>10</sup></b> <b>Networked Smart Learning Objects for Online Laboratories Working Group</b>	2012	É um projeto, denominado P1876 - <i>Networked Smart Learning Objects for Online Laboratories</i> , suportado pela associação IEEE para uniformizar os laboratórios acedidos remotamente. O principal objetivo desta norma é estabelecer a relação entre todos os componentes envolvidos num cenário de experimentação remota (software, hardware e sistemas de aprendizagem), a fim de facilitar a conceção e implementação destas experiências no desenvolvimento de ambientes de aprendizagem [1].
<b>RExLab<sup>11</sup></b> <b>Laboratório de Experimentação Remota</b>	1997	Tem como objetivo a partilha de recursos para serem concedidos a quem mais necessita deles (alunos, professores, instituições de ensino, entre outros). Evoluiu para a formação de uma Rede de Experimentação Remota denominada RExNet, distribuída em seis países diferentes, e constituída por doze faculdades. Todas as instituições fornecem mão-de-obra qualificada com o objetivo de melhorar o nível educacional, acentuando atividades que estimulem a criatividade, o interesse e a experimentação remota [13].
<b>Virtual Labs<sup>12</sup></b>	2012	É um projeto iniciado pelo Ministério do Desenvolvimento de Recursos Humanos do Governo da Índia, sob a Missão Nacional de Educação através das TIC. Este projeto foi criado com os seguintes objetivos: i) disponibilizar o acesso remoto a laboratórios de várias disciplinas das Ciências e Engenharias, ii) despertar os alunos para a realização de experiências, iii) ajudar os alunos na aprendizagem de conceitos básicos e avançados através da experimentação remota, iv) proporcionar um completo Sistema de Gestão de Aprendizagem em torno dos Laboratórios Virtuais e, v) partilhar equipamentos e recursos a qualquer hora e em qualquer lugar.
<b>VISIR<sup>13</sup></b> <b>Virtual Instrument Systems in Reality</b>	1999	É um laboratório remoto para a ligação e medição de circuitos eletrónicos, disponível para alunos e professores. Num laboratório VISIR, as mãos humanas são substituídas por telemanipuladores que executam as ações manuais necessárias. Cada utilizador pode construir virtualmente a sua rede de circuitos que, antes de ser executada, é validada para proteger todos os equipamentos reais utilizados. O <i>VISIR Open Lab Platform (5.0)</i> é uma arquitetura que pretende a união de vários laboratórios VISIR e, além disso, incluir um repositório partilhado de material de aprendizagem [17].

Ao longo dos últimos anos, com a criação de várias infraestruturas de laboratórios remotos foram também surgindo problemas devido à sua diversificação. Estes problemas, embora sejam debatidos por alguns dos projetos descritos anteriormente, ainda não têm uma solução uniforme.

<sup>10</sup> Projeto disponível em <http://ieee-sa.centraldesktop.com/1876public/>

<sup>11</sup> Projeto disponível em <http://www.rexlab.ufsc.br/?q=pt-br/home>

<sup>12</sup> Projeto disponível em <http://www.vlab.co.in/>

<sup>13</sup> Projeto disponível em <http://visir.ieec.uned.es>

## 2.5 Problemas atuais e solução proposta

Atualmente as experiências remotas são acedidas através de interfaces 2D, e mais recentemente, a utilização de interfaces 3D está a ser considerada [3], uma vez que torna o ambiente mais intuitivo, aproximando os laboratórios remotos dos tradicionais permitindo um aumento do interesse e da motivação dos seus utilizadores.

A evolução tecnológica tem contribuído para o crescimento dos laboratórios remotos, ao criarem equipamentos cada vez mais sofisticados, providos com interfaces de rede para serem mais facilmente integrados nas atuais infraestruturas. Este crescimento é notório pelo número de laboratórios remotos criados pelas instituições de ensino.

No entanto, com o crescimento dos laboratórios remotos surgiram também alguns problemas de normalização. Como a infraestrutura de cada laboratório remoto é desenvolvida segundo um critério específico e distinto, foram utilizadas ao longo dos anos várias arquiteturas de *hardware* e *software*, cada uma com diferentes linguagens de programação.

A inexistência de uma norma para o desenvolvimento de laboratórios remotos dificulta a sua disseminação, criando assim os seguintes problemas [3]:

- Não promove a colaboração entre as instituições porque é difícil a reutilização e ligação de diferentes módulos de laboratório de experiências específicas;
- Dificulta a criação de laboratórios remotos, na medida que necessita de certos conhecimentos técnicos que apenas são encontrados nos cursos de engenharia;
- Custos avultados para a criação de uma infraestrutura laboratorial remota, devido à necessidade de um computador que funcione como servidor, de aplicações de *software* para gerir todo o sistema, e de equipamentos para serem utilizados;
- Limita a realização de diferentes experiências quando a arquitetura de um laboratório remoto é baseado apenas num único computador;
- Dificulta a atualização do *software* utilizado, devido aos problemas de não compatibilidade entre as diferentes versões.

Para superar estes problemas, foi desenvolvido no âmbito de um trabalho de doutoramento [1] um protótipo que propõe a criação de laboratórios remotos reconfiguráveis baseados na programação de um dispositivo lógico programável denominado FPGA (*Field Programmable Gate Arrays*). Estes dispositivos estão tipicamente disponíveis em cartas possuindo uma vasta gama de periféricos tais como: conversores digital-analógico (*Digital-to-Analog Converter*, DAC), conversores analógico-digital (*Analog-to-Digital Converter*, ADC), portas de interface, entre outros. Estes podem ser interligados às experiências. Uma solução deste género permite embutir os vários módulos de laboratório dentro dos seus blocos de memória, não sendo necessário utilizar o equipamento tradicional de um laboratório remoto (computador e os vários equipamentos independentes). Além disso, os módulos de laboratório são descritos através de uma linguagem normalizada para *hardware* (*Hardware Description Language*, HDL) e a sua estrutura é definida segundo a norma IEEE 1451.0, que será resumidamente descrita

no capítulo seguinte por se tratar de um tema importante para o trabalho desenvolvimento nesta dissertação [2][5][6].

## **2.6 Sumário**

Neste capítulo foram descritos os vários objetivos da experimentação remota. Explicaram-se as vantagens e desvantagens dos laboratórios remotos no ensino, tendo em consideração os seus principais atores (alunos, professores e as instituições de ensino). Foi descrita a arquitetura de um laboratório remoto e as características da respetiva infraestrutura que a suportam. Para completar esta descrição foi realizado um levantamento dos projetos e consórcios que contribuem para a evolução dos laboratórios remotos, referindo para cada um deles quais as suas áreas de competência. Concluiu-se o capítulo com a enumeração dos atuais problemas na implementação dos laboratórios remotos e com a apresentação de uma solução que possibilita a reconfiguração da infraestrutura subjacente com diferentes módulos, cuja descrição é efetuada no capítulo seguinte.





## 3 A norma IEEE 1451.0 no desenvolvimento de laboratórios remotos

Atualmente o crescente desenvolvimento de laboratórios remotos está condicionado pela inexistência de uma norma que regularize as suas arquiteturas. Para a resolução deste problema, é apresentado um protótipo idealizado e desenvolvido no âmbito de um trabalho de doutoramento [1], cuja arquitetura tem por base a norma IEEE 1451.0. Este protótipo é suportado por uma infraestrutura reconfigurável implementada com base em FPGAs<sup>14</sup>, cuja capacidade de reconfiguração permite a utilização de módulos de laboratório embutidos para a monitorização das experiências remotas.

Neste capítulo descreve-se a família de normas IEEE 1451 destacando-se a norma IEEE 1451.0 e a arquitetura associada, composta por um processador de rede (*Network Capable Application Processor*, NCAP) interligado a um módulo para a ligação de transdutores inteligentes (*Transducer Interface Module*, TIM) que, no âmbito do protótipo desenvolvido, são os módulos de laboratório a utilizar num laboratório remoto. De seguida, dá-se uma particular atenção a um conjunto de memórias denominadas por TEDSs (*Transducer Electronic Data Sheets*) por se tratarem do tema central do trabalho desenvolvido. Conclui-se o capítulo descrevendo-se genericamente a arquitetura de um laboratório remoto reconfigurável e o enquadramento do trabalho desenvolvido, cuja descrição é efetuada no capítulo seguinte.

### 3.1 A família de normas IEEE 1451.x

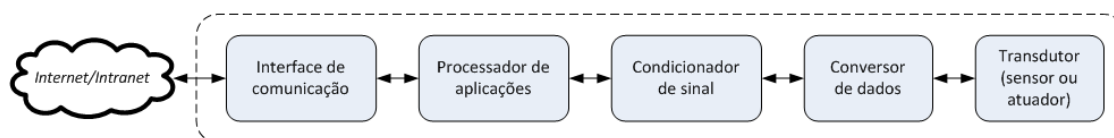
A família de normas IEEE 1451.x define um conjunto de interfaces de comunicação que permitem a interligação em rede de transdutores inteligentes a sistemas com microprocessadores. A criação destas normas pelo *National Institute of Standards and*

---

<sup>14</sup> <http://reliability.ee.byu.edu/>

Technology (NIST)<sup>15</sup> em colaboração com a organização *Institute of Electrical and Electronics Engineers (IEEE)*<sup>16</sup> uniformizou a utilização dos transdutores inteligentes e a sua disseminação em diferentes domínios [19][20]. Neste contexto, o desenvolvimento destas normas permitiu que os transdutores inteligentes, no momento em que são ligados em rede, tenham a capacidade de se identificarem e de se autoconfigurarem no sistema sem a necessidade de intervenção humana, ficando capacitados de processarem os dados recolhidos do exterior e de os transmitir em rede através de protocolos de comunicação.

Um transdutor é um sensor ou um atuador. O sensor é responsável por gerar um sinal elétrico correspondente a um dado físico, biológico ou químico, adquirido do meio onde está localizado. O atuador, após receber um sinal elétrico, é responsável por gerar uma ação física no meio onde está inserido. Um transdutor torna-se inteligente quando é interligado por cinco partes representadas na figura 3.1 [7][21]: i) transdutor (sensor ou atuador), ii) conversor de dados (conversor do sinal), iii) condicionador de sinal (para amplificar, filtrar e equalizar o sinal, com o objetivo de garantir uma boa relação sinal/ruído e uma menor distorção harmónica), iv) processador das aplicações e, v) interface de comunicação com a rede.



**Figura 3.1: Modelo de um transdutor inteligente.**

Os dados recolhidos por um sensor analógico são amplificados, filtrados e convertidos num sinal digital utilizando um ADC, sendo posteriormente processados por um microprocessador. Os dados são depois enviados para um computador ou para um sistema de monitorização através da *Internet/Intranet*. A monitorização de um atuador é realizada de forma inversa ao sensor. Este é controlado através de uma ação enviada por um computador na rede que, depois de convertida, fornece as instruções ao atuador.

A arquitetura de um transdutor inteligente, segundo as normas da família IEEE 1451.x, divide o modelo apresentado em dois componentes: processador e transdutor, ambos interligados por uma interface física. Para além desta divisão, é ainda adicionada uma estrutura de dados eletrónicos denominada por TEDS (*Transducer Electronic Data Sheets*) que armazena a informação relacionada com os transdutores. Ambos os componentes são definidos de forma distinta pela família de normas IEEE 1451.x, que atualmente é composta por seis normas ativas, uma obsoleta e uma em desenvolvimento, conforme é representado na tabela 3.1 [21].

<sup>15</sup> <http://www.nist.gov/>

<sup>16</sup> <http://www.ieee.org>

**Tabela 3.1: Família de normas IEEE 1451.x.**

<b>Norma</b>	<b>Data</b>	<b>Objetivo</b>
IEEE 1451.0	2007	Define um conjunto de funcionalidades comuns, comandos e TEDSs, para serem usados pela família de normas IEEE 1451.x. Inclui as funções básicas para controlar e gerir transdutores inteligentes, protocolos de comunicação e os comandos de leitura e escrita dos TEDSs e das suas respetivas configurações. Define ainda um conjunto de Interfaces de Programação de Aplicações ( <i>Application Programming Interface</i> , API) para facilitar a comunicação com o transdutor e as aplicações [22].
IEEE 1451.1	1999	Define o modelo geral e as especificações das interfaces para ligar o processador das aplicações com a rede de comunicação e com os transdutores. Um dos seus princípios é a definição das comunicações entre os processadores. Atualmente está desatualizada [22][23].
IEEE 1451.2	1997	Define a interface entre o transdutor e o processador, e os TEDSs para uma comunicação ponto-a-ponto <sup>17</sup> (e.g. porta paralela DB-25, porta RS232, <i>Universal Serial Bus</i> (USB)). A ligação física é feita através de uma interface digital de 10 fios, também conhecida por <i>Transducer Independent Interface</i> (TII). Está a ser atualizada para ser utilizada em conjunto com a norma IEEE 1451.0, e para possibilitar mais dois tipos de interfaces de comunicação: <i>Universal Asynchronous Receiver/Transmitter</i> (UART) e <i>Universal Serial Interface</i> (USI) [19][22].
IEEE 1451.3	2003	Define a interface entre o transdutor e o processador, e os TEDSs para utilizarem um protocolo de comunicação multiponto <sup>18</sup> . Está a ser revista para ser utilizada em conjunto com a norma IEEE 1451.0 [22][24].
IEEE 1451.4	2004	Define uma interface mista ( <i>Mixed-Mode Interface</i> , MMI), onde pelo mesmo meio físico é possível ter uma interface analógica para o sinal do sensor, e um canal digital de baixo custo para comunicar com os TEDSs. A utilização de sensores analógicos (e.g. acelerómetros) em conjunto com a norma IEEE 1451.0 obriga a que os TEDSs definidos pela norma IEEE 1451.4 sejam convertidos para a norma IEEE 1451.0 [7][22].
IEEE 1451.5	2007	Define a interface entre o transdutor e o processador, e os TEDSs para transdutores sem fios (e.g. 802.11 ( <i>WiFi</i> ), 802.15.1 ( <i>Bluetooth</i> ), 802.15.4 ( <i>ZigBee</i> ) e <i>Low power Wireless Personal Area Networks</i> (6LowPAN)). Um NCAP pode comunicar com um ou mais <i>Wireless Transducer Interface Module</i> (WTIM), desde que o transdutor esteja ao alcance da sua rede sem fio [22][25].
P1451.6	-	Trata-se de uma proposta ainda em desenvolvimento que define a interface entre o transdutor e o processador, e os TEDSs utilizados numa rede de alta velocidade denominada por CAN [24][20].
IEEE 1451.7	2010	Define as interfaces e o protocolo de comunicação entre os transdutores e os sistemas de identificação por radiofrequência ( <i>Radio-Frequency IDentification</i> , RFID) [24].

A arquitetura do protótipo desenvolvido, descrito neste capítulo, tem por base a norma IEEE 1451.0. Desta forma, a norma IEEE 1451.0 será brevemente descrita com particular enfoque na definição e estrutura dos TEDSs.

<sup>17</sup> A comunicação ponto-a-ponto é uma arquitectura de redes de computadores onde cada um dos nós pode funcionar como cliente ou servidor, permitindo partilhar recursos e informações da rede.

<sup>18</sup> O sistema *Multidrop Bus* (MDB) permite a ligação de vários dispositivos (e.g. transdutores) através de um único canal de comunicação.

### 3.2 A norma IEEE 1451.0: arquitetura e funcionalidades

A norma IEEE 1451.0 foi desenvolvida com o objetivo de uniformizar a família de normas IEEE 1451.x. Esta norma define um conjunto de funções para aceder a transdutores e simplifica a criação de futuras normas que estabeleçam a interligação física entre o NCAP e o TIM. A figura 3.2 ilustra a relação entre a norma IEEE 1451.0 e as restantes normas da família IEEE 1451.x, representando as diferentes normas associadas aos dois principais componentes definidos na arquitetura: o NCAP e o TIM, e a interligação destes com os TEDSs e os restantes módulos de laboratório [21][22].

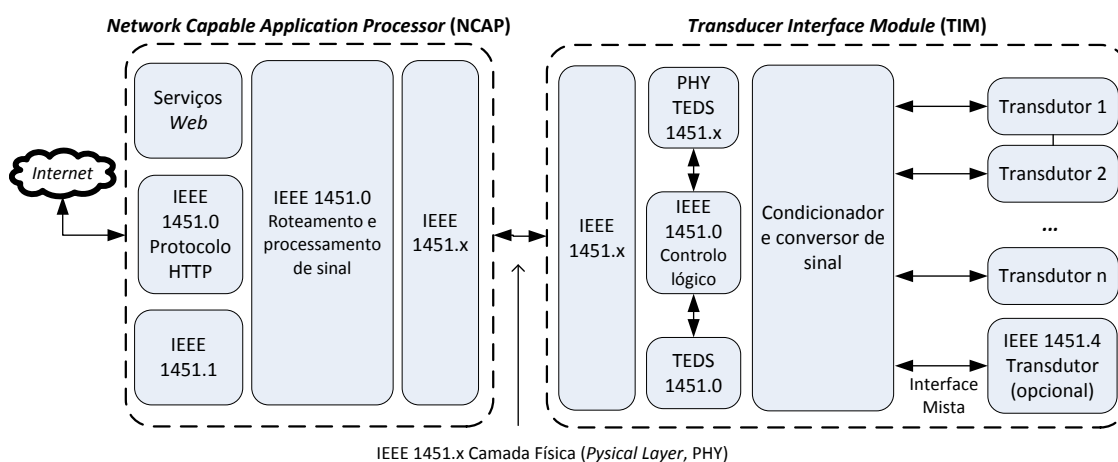


Figura 3.2: Arquitetura da norma IEEE 1451.0.

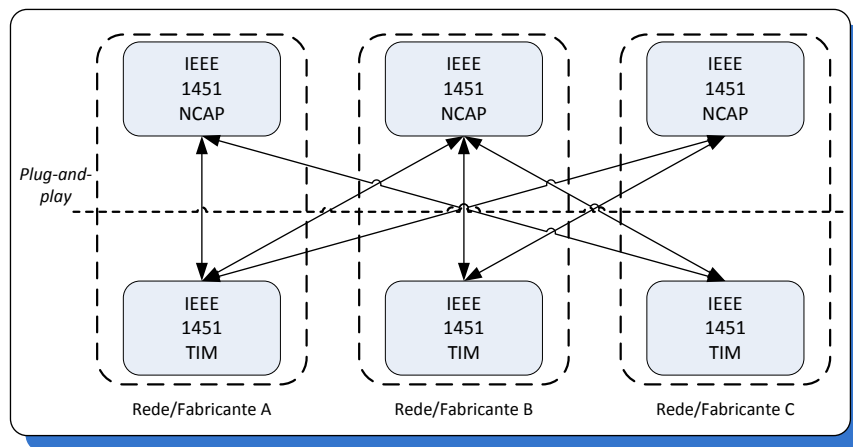
O TIM contém a interface lógica do sensor/atuador. Este é responsável pelas operações de sinal (amplifica e filtra o sinal) e pela conversão de dados. Por regra, o TIM é responsável por guardar os TEDSs, que contêm a calibração e a informação necessária para converter os dados lidos ou para a escrever pelo transdutor no Sistema Internacional de Unidades (SI). Estes dados são transmitidos entre o NCAP e o TIM através de uma interface física regulada por uma norma da família IEEE 1451.x [16][26]. Existem três formas para aceder aos sensores/atuadores interligados no TIM a partir da rede: i) utilizando o protocolo HTTP (*Hypertext Transfer Protocol*) especificado na norma IEEE 1451.0, ii) pelo modelo de comunicação definido na norma IEEE 1451.1 ou, iii) através de serviços *Web*<sup>19</sup> [21][22].

O NCAP tem a responsabilidade de gerir os vários TIMs e de processar todos os dados usados numa dada aplicação informática. Quando o NCAP é inicializado, faz uma pesquisa por todos os TIMs que estão ligados às suas interfaces. Por regra, no momento em que são encontrados, copia cada um dos TEDSs, que estão guardados em cada TIM, para dentro de uma memória no NCAP. O NCAP tem ainda a função de corrigir os dados recebidos pelo TIM, convertendo-os nas unidades SI definidos pelos TEDSs. No final, os dados são transferidos para uma rede externa, usando o protocolo HTTP e XML [21][22].

<sup>19</sup> Um serviço *Web* é um conjunto de serviços disponíveis na *Internet* através de uma interface XML, permitindo a comunicação de diferentes aplicações, independentemente do sistema operativo e da linguagem de programação.

A interface física entre o NCAP e o TIM é definida pelas normas IEEE 1451.2 (ponto-a-ponto), IEEE 1451.3 (sistema multiponto), IEEE 1451.5 (sem fio), IEEE P1451.6 (CAN) e IEEE 1451.7 (identificação por radiofrequência). Por regra, as interfaces entre os transdutores e o local onde é realizado o condicionamento de sinal e a conversão dos dados, não são definidas pela norma IEEE 1451, com a exceção da norma IEEE 1451.4, que especifica uma interface de baixo nível para os transdutores [7].

A norma IEEE 1451.0 possibilita que diferentes fabricantes produzam os seus transdutores com a garantia de que serão configurados automaticamente sem ocorrerem problemas no momento de ligação com os equipamentos de outros fabricantes (funcionalidade *plug-and-play*), como é ilustrado pela figura 3.3 [21].



**Figura 3.3: Capacidade *plug-and-play* dos módulos dos transdutores.**

A capacidade *plug-and-play* dos transdutores permitem concluir que:

- Os TIMs de diferentes fabricantes podem ser ligados, através do mesmo módulo de comunicação, a NCAPs de diferentes redes de transdutores;
- Os NCAPs podem ser ligados a um grande número de TIMs através da interface física definida pela família de normas IEEE 1451.x. Os TIMs podem ser de diferentes fabricantes e de outras redes.

Desta forma, o TIM, no momento de ligação a um sistema desconhecido, é configurado e reconhecido pelo NCAP, sem haver um qualquer conhecimento prévio de ambos. Para garantir essa funcionalidade é necessário a definição da estrutura de dados dos TEDSs.

### 3.3 Estrutura de dados: TEDS

Uma das características fundamentais de um transdutor inteligente definido segundo a norma IEEE 1451.0 é a especificação dos TEDSs<sup>20</sup>. Os TEDSs são estruturas de dados que armazenam informação relacionada com os transdutores, tais como: a identificação, a identificação do fabricante, o tipo de transdutor (sensor ou atuador), os canais do transdutor

<sup>20</sup> No anexo A são detalhadas as estruturas de dados de vários TEDSs definidos na norma IEEE 1451.0.

(*TransducerChannels*<sup>21</sup>, TCs), número de série, o intervalo de medição, os dados de calibração, entre outros [22]. Por regra, os TEDSs estão localizados no TIM. No momento em que o TIM é identificado pelo NCAP é transferido uma cópia de cada TEDS para a sua memória, se a informação for apenas de leitura. Os TEDSs são então armazenados no NCAP numa memória interna apagável eletricamente, programável, exclusivamente de leitura (*Electrically-Erasable Programmable Read-Only Memory*, EEPROM). No caso da informação dos TEDSs serem de leitura e escrita os dados são guardados numa memória de acesso aleatório (*Random Access Memory*, RAM). Outra possibilidade é a utilização de TEDSs virtuais, que são armazenados permanentemente numa memória fora do TIM (e.g. num repositório remoto acedido através de um URL) [7][22]. A norma IEEE 1451.0 definiu quatro TEDSs obrigatórios e vários TEDSs opcionais, que são descritos na tabela 3.2 [22].

**Tabela 3.2: Tipos de TEDSs definidos pela norma IEEE 1451.0.**

	<b>Nome do TEDS</b>	<b>Descrição</b>
Obrigatórios	<b>Meta-TEDS</b>	Define as informações necessárias para aceder a um qualquer <i>TransducerChannel</i> . Define ainda um identificador único e universal ( <i>Universally Unique Identifier</i> , UUID) do TIM e especifica os intervalos de tempo para que o NCAP considere, no pior caso, um TIM indisponível.
	<b>TransducerChannel TEDS</b>	Descreve um conjunto de informações específicas do transdutor, tais como: o tipo de transdutor, o parâmetro físico medido ou controlado, o modo de funcionamento e o intervalo de tempo em que deve operar.
	<b>User's Transducer Name TEDS</b>	Armazena o nome pelo qual o sistema e o utilizador reconhecem o transdutor.
	<b>PHY TEDS</b>	Indica a camada física que permite a ligação do NCAP com o TIM. Apesar dos métodos de acesso serem descritos pela norma IEEE 1451.0, a estrutura do TEDS e a sua implementação são definidas pelas restantes normas da família IEEE 1451.
Opcionais	<b>Calibration TEDS</b>	Disponibiliza toda a calibração necessária para os dados serem convertidos à entrada e à saída do transdutor.
	<b>Frequency Response TEDS</b>	Fornece os dados sobre a frequência de resposta dos TCs ao utilizador.
	<b>Transfer Function TEDS</b>	Descreve uma forma para ligar em conjunto uma série de funções individuais de transferência para descrever a frequência de resposta dos TCs de forma algorítmica.
	<b>Text-based TEDS</b>	Os TEDSs baseados em texto ( <i>Commands TEDS</i> , <i>Identification TEDS</i> , <i>Geographic location TEDS</i> e <i>Units extension TEDS</i> ) fornecem a informação em texto sobre o TIM ou os TCs.
	<b>End User Application Specific TEDS</b>	Disponibiliza um bloco de memória onde o utilizador pode definir um valor variável, em qualquer formato, para ser armazenado no TIM.
	<b>Manufacturer-defined TEDS</b>	A informação descrita neste TEDS é da responsabilidade do fabricante e pode estar guardada em qualquer formato. Este TEDS não necessita de estar disponível para os utilizadores.

<sup>21</sup> Um *TransducerChannel* é o nome dado ao conjunto formado pelo transdutor e o condicionador e conversor de sinal. Cada TC tem um número de identificação de 16 bits, atribuído pelo TIM. Funciona ainda como endereço de destino dos comandos enviados pelo NCAP para o TIM [22].

### 3.3.1 Vantagens dos TEDSs

Os TEDSs são essenciais para o funcionamento independente dos transdutores, fornecendo a sua identificação e as especificações detalhadas de cada um. A definição dos TEDSs proporciona benefícios aos transdutores [7] [21]:

- Permitem a identificação automática dos transdutores: qualquer transdutor que defina os TEDSs segundo a norma IEEE 1451.0 tem a capacidade para se identificar e descrever na rede, enviando para o NCAP as informações guardadas no TEDS;
- Armazenam dados: os TEDSs de um transdutor podem guardar a informação sobre a sua localização, calibração, registos de reparação e outros dados relacionados com a sua manutenção, que podem ser utilizados para gerar documentação a longo prazo;
- Automatizam operações: como a informação guardada nos TEDSs é transferida automaticamente na rede, elimina-se a necessidade da introdução manual dos parâmetros nos transdutores, reduzindo dessa forma o erro humano;
- Facilitam a instalação, atualização e manutenção dos transdutores: reduzem o custo total do ciclo de vida dos sistemas associados;
- Fornecem a capacidade de *plug-and-play*: o TIM e o NCAP construídos e conectados com base na norma IEEE 1451.0 têm a capacidade de se identificarem e comunicarem sem a necessidade de qualquer intervenção humana ou alterações no *software*.

### 3.3.2 Estrutura de um TEDS

Os TEDSs definidos pela norma IEEE 1451.0 utilizam o formato genérico, ilustrado pela tabela 3.3 [22][27].

Tabela 3.3: Formato genérico de um TEDS.

Campo	Descrição	Tipo	N Octetos <sup>22</sup>
-	Tamanho do TEDS	UInt32 <sup>23</sup>	4
1-N	Estrutura de dados	Variável	Variável
-	<i>Checksum</i>	UInt16 <sup>24</sup>	2

O primeiro campo representa o número de octetos (*TEDS length*), definido em quatro octetos através de um valor inteiro positivo, de toda a estrutura do TEDS, ao qual são adicionados dois octetos, do campo *checksum*.

O segundo campo contempla uma estrutura de dados (*Data block*) com toda a informação guardada num TEDS. Esta pode ser binária ou baseada em texto, pode variar de tamanho consoante o TEDS a especificar e deverá ser descrita utilizando uma estrutura denominada TLV dividida em três campos: *Tipo/Tamanho/Valor* (*Type/Length/Value*).

<sup>22</sup> Um octeto, conhecido como *byte*, é um grupo de oito *bits*.

<sup>23</sup> UInt32 = Inteiro de 32 *bits* (valor entre -2 147 483 648 e 2 147 483 647).

<sup>24</sup> UInt16 = Inteiro de 16 *bits* (valor entre zero e 65 535).

Numa estrutura TLV os dados guardados pelos TEDSs são definidos por um tuplo, cujos campos têm uma função específica:

- *Tipo*: responsável por identificar o tuplo através de um identificador único;
- *Tamanho*: representa a soma de todos os octetos do campo *Valor*;
- *Valor*: guarda a informação de cada tuplo. Pode ser composto por um ou mais subtuplos, como é representado na tabela 3.4.

**Tabela 3.4: Exemplo de uma estrutura TLV.**

Número de tuplos	Estrutura TLV	Número de octetos	
1 tuplo simples	<i>Tipo</i>	13 (Identificador único)	
	<i>Tamanho</i>	02 (Soma do campo <i>Valor</i> - octetos)	
	<i>Valor</i>	00.02	
1 tuplo composto por subtuplos	<i>Tipo</i>	14 (Identificador único)	
	<i>Tamanho</i>	11 (Soma do campo <i>Valor</i> - octetos)	
	<i>Valor</i>	<i>Tipo</i>	20
		<i>Tamanho</i>	01
		<i>Valor</i>	05
	<i>Valor</i>	<i>Tipo</i>	21
		<i>Tamanho</i>	02
		<i>Valor</i>	00.06
		<i>Tipo</i>	21
		<i>Tamanho</i>	02
		<i>Valor</i>	00.10

Na tabela é possível verificar que para o tuplo simples com o identificador único (13) apenas existe um campo *Valor* (00.02), cuja soma dos seus octetos é definida no campo *Tamanho* (02). No caso de um tuplo composto por vários subtuplos a estrutura TLV é igualmente respeitada. O tuplo com o identificador único (14) tem um campo *Valor* composto por três subtuplos simples: [20.01.05], [21.02.00.06] e [21.02.00.10], cada um definido segundo a estrutura TLV. O cálculo da soma de todos os octetos definidos no campo *Valor*, do tuplo com o identificador único (14), é guardado pelo campo *Tamanho* (11).

O terceiro campo da estrutura genérica de um TEDS é reservado para o *checksum*. Este é utilizado para verificar e garantir a integridade de cada TEDS. O *checksum* é calculado efetuando o complemento para um da soma de todos os octetos anteriores, incluindo o campo *Tamanho* do TEDS e de toda a estrutura de dados associada, excluindo o próprio campo *checksum*, conforme a seguinte fórmula [22]:



$$checksum = 0xFFFF - \sum_{i=1}^{\text{TotalOctetos}-2} TEDS\ Octeto(i) \quad (1)$$

Os tuplos definidos para cada TEDS variam consoante as suas funções. No entanto, existe um tuplo obrigatório para todos os TEDSs denominado por TEDS *Identification Header*. Esta estrutura é composta por quatro campos (*Família, Classe, Versão, Tamanho do Tuplo*), como é demonstrado na tabela 3.5 [22]. Este tuplo tem o objetivo de descrever cada um dos TEDSs, sendo também o primeiro tuplo a ser definido na sua estrutura de dados.

**Tabela 3.5: Estrutura do TEDS *Identification Header*.**

Campo	Valor	Função
<b>Tipo</b>	03	Identificador único da estrutura de dados <i>Identification Header</i> . O seu valor é sempre três.
<b>Tamanho</b>	04	Tamanho da estrutura de dados. O valor é sempre igual a quatro, dado que o <i>Identification Header</i> tem quatro octetos.
<b>Família</b>	00	Identifica qual das normas da família IEEE 1451 é que o TEDS está associado. Se for associado à norma IEEE 1451.0 o seu valor deve ser zero.
<b>Classe</b>	—	Identifica o TEDS acedido através do seu código de acesso.
<b>Versão</b>	—	Identifica a versão do TEDS.
<b>Tamanho do Tuplo</b>	01	Identifica o número de octetos do campo <i>Tamanho</i> de todos os TLVs do TEDS, com a exceção deste. Por regra é atribuído o valor de um, o que significa que existem no máximo 255 octetos no campo <i>Valor</i> .

A estrutura de dados de cada TEDS e a arquitetura da norma IEEE 1451.0 foram a solução seguida na adopção dos módulos de laboratório ao serem utilizados pelo protótipo de laboratórios remotos apresentado na subsecção seguinte.

### 3.4 Exemplo de um laboratório remoto reconfigurável baseado na norma IEEE 1451.0

Os laboratórios remotos tradicionais são um importante meio para os diferentes atores envolvidos no processo do ensino e aprendizagem. No entanto, não permitem a alteração ou definição da instrumentação necessária para a realização de experiências remotas. Estas limitações impedem o desenvolvimento de algumas competências consideradas fundamentais no ensino da engenharia, nomeadamente a estruturação e construção de instrumentos e sistemas, conforme sugere a organização ABET<sup>25</sup> (*Accreditation Board for Engineering and Technology*) [5]. Atualmente existem vários laboratórios remotos reconfiguráveis. Todavia, as suas infraestruturas apresentam ainda várias limitações, não permitindo a seleção ou alteração dos módulos de laboratório, tal como num laboratório tradicional [5].

Durante vários anos cada instituição de ensino construiu o seu laboratório remoto implementando a sua própria arquitetura, seguindo uma filosofia não normalizada [3]. Esta

<sup>25</sup> <http://www.abet.org/>

situação dificultou a reutilização dos diferentes módulos de laboratório utilizados nas experiências disponibilizadas por diversas instituições, impossibilitando a sua troca e tornando ainda mais complexo o acesso às experiências. Além disso, é comum a existência de um elevado custo na implementação das infraestruturas laboratoriais remotas, pois estas são construídas com base em vários equipamentos (servidor, *software* de controlo, instrumentos, entre outros).

Para contribuir para a resolução destes problemas e com o objetivo de difundir a utilização dos laboratórios remotos no ensino considerou-se importante que as infraestruturas possam: i) ser reconfiguradas com diferentes módulos de laboratório e ii) garantir a sua implementação e acesso de forma normalizada. Apesar de atualmente existirem vários projetos e consórcios a debater os problemas mencionados, nomeadamente o comité técnico do GOLC, ainda não existe uma solução para os resolver [2][6]. É neste cenário que um laboratório remoto reconfigurável e normalizado pode contribuir para superar os problemas mencionados. Desta forma, o laboratório remoto que serve de base ao trabalho descrito nesta dissertação permite a especificação e preparação das experiências através da seleção dos módulos de laboratório de forma semelhante à efetuada num laboratório tradicional. Importa, por isso, fazer uma breve descrição da sua arquitetura e apresentar alguns aspetos subjacentes ao seu desenvolvimento.

### 3.4.1 Arquitetura adotada

Conforme ilustrado na figura 3.4, a arquitetura do laboratório existente é composta por três partes: i) atores, que representam os utilizadores do laboratório, ii) servidor, que fornece a interface remota de acesso ao laboratório e que permite a importação de ficheiros com a configuração dos módulos de laboratório e, iii) infraestrutura laboratorial remota, que permite a interligação dos atores às experiências disponibilizadas [6].

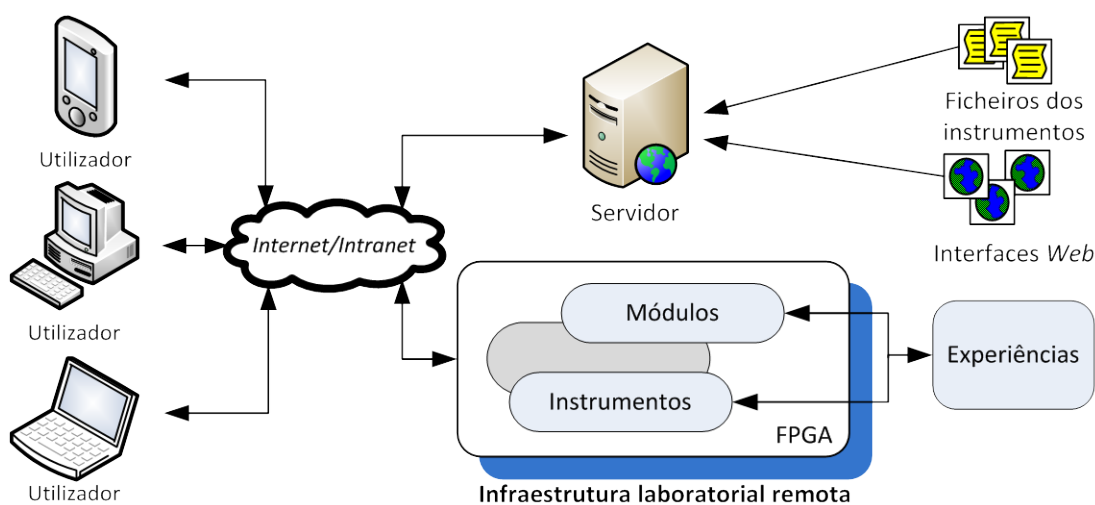


Figura 3.4: Arquitetura do protótipo do laboratório remoto reconfigurável.

O laboratório remoto foi desenvolvido com base na norma IEEE 1451.0 e permite a reconfiguração da infraestrutura subjacente recorrendo à utilização de um dispositivo lógico programável, nomeadamente de uma FPGA, para a integração de vários módulos de laboratório [2][6]. Para o controlo normalizado dos módulos de laboratório, os utilizadores utilizam uma API HTTP definida pela norma IEEE 1451.0. A descrição dos módulos de laboratório com base na norma IEEE 1451.0 garante a sua facilidade de criação e controlo, contribuindo assim para uma solução universal [3][5].

Na infraestrutura laboratorial remota a FPGA agrupa os diferentes módulos de laboratório nos seus blocos de memória para a realização de uma determinada experiência. Devido às suas capacidades de reconfiguração, a utilização dos instrumentos tradicionais existentes num laboratório remoto torna-se assim desnecessária, reduzindo dessa forma os custos associados [3]. O protótipo descrito faz ainda uso de linguagens normalizadas para definir os módulos de laboratório, mais concretamente, através de ficheiros HDL (e.g. VHDL/Verilog). Desta forma, é garantida a interligação dos módulos de laboratório de forma independente do fabricante da FPGA [3][5].

A partilha dos módulos de laboratório entre diferentes instituições e a respetiva reconfiguração de uma FPGA é efetuada através do uso de um servidor *Web*. Para o controlo e gestão remota de cada módulo é necessário a implementação e disponibilização de interfaces *Web*. Desta forma, sem a necessidade de grandes conhecimentos técnicos, as instituições podem criar os seus laboratórios remotos reutilizando os vários recursos existentes na *Internet*, promovendo assim a cooperação entre as diversas instituições e possibilitando a realização de atividades em grupos de trabalhos, tal como num laboratório tradicional [3].

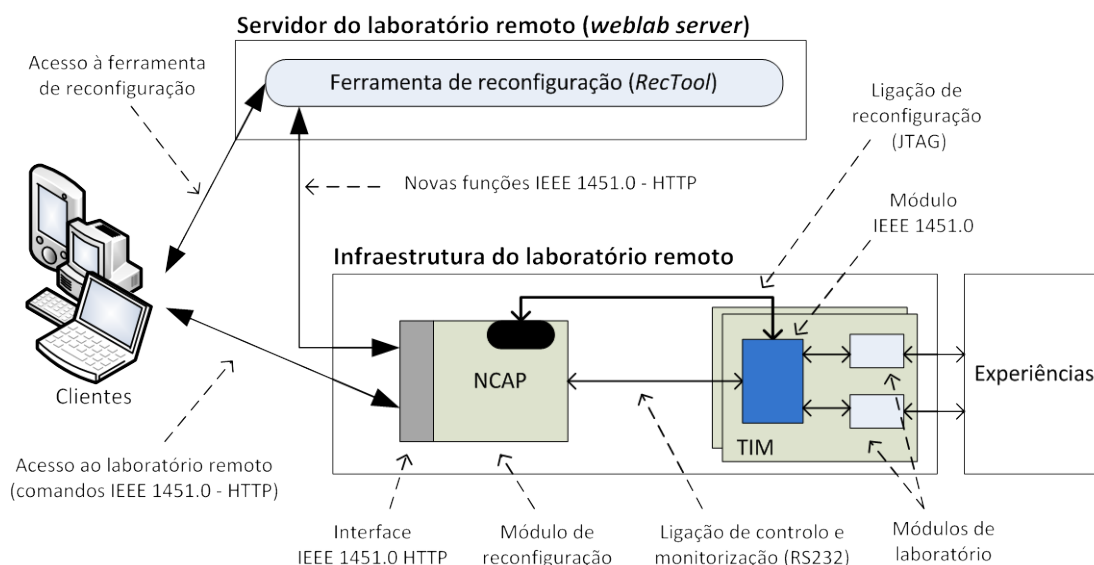
### 3.4.2 Aspetos de desenvolvimento

O laboratório remoto reconfigurável é definido por um servidor, denominado *weblab server*, implementado através de um simples computador e por uma infraestrutura que tem por base o modelo de referência definido na norma IEEE 1451.0 composta pela interligação entre o NCAP e o TIM, como é ilustrado pela figura 3.5 [5].

Na infraestrutura laboratorial remota o TIM é implementado pela FPGA, que armazena os módulos de laboratório, cuja responsabilidade é a de comunicar com as várias experiências. O NCAP é constituído por um microcomputador, que exerce também a função de um pequeno servidor *Web*, acedido através de comandos HTTP definidos na norma IEEE 1451.0. A interligação entre o NCAP e o TIM é realizada através de dois tipos de ligação: i) reconfiguração, implementada por um barramento *Joint Test Action Group* (JTAG<sup>26</sup>) utilizada para reconfigurar a FPGA e, ii) controlo e monitorização, por meio de um barramento RS-232, que permite aos utilizadores a gestão remota de cada módulo de laboratório embutido na FPGA, usando os comandos HTTP definidos pela norma IEEE 1451.0 [2][5].

---

<sup>26</sup> <http://standards.ieee.org/findstds/standard/1149.1-1990.html>



**Figura 3.5: Diagrama de blocos da arquitetura do laboratório remoto.**

A interação com o servidor do laboratório remoto é feita através de uma ferramenta de reconfiguração (*Reconfiguration Tool, RecTool*), usando um *Web browser*, sem a necessidade de instalar novas extensões ou *softwares* específicos nos dispositivos dos utilizadores. A *RecTool* permite a reconfiguração da infraestrutura do laboratório com diferentes módulos de laboratório, para a realização de uma determinada experiência. Esta ferramenta fornece vários métodos para criar ficheiros em código binário, usados na reconfiguração do TIM. Os ficheiros são gerados após a definição ou seleção dos módulos de laboratório pelo utilizador. De seguida, cada módulo é embutido na FPGA juntamente com um módulo IEEE 1451.0 predefinido, descrito por um ficheiro HDL, de acordo com a norma IEEE 1451.0 [3][5].

### 3.4.3 Funcionalidades e interface da ferramenta de reconfiguração

A *RecTool* é acedida através da interface ilustrada na figura 3.6. Esta interface permite aos atores o controlo do processo de reconfiguração através da submissão de ficheiros de *Configuração* e de *Projeto* do laboratório remoto. Para facilitar a apresentação da informação do processo de reconfiguração, a interface foi dividida em três painéis: *Build*, *Reconfiguration* e *Reports* [4][5].

No painel *Reconfiguration* estão guardados todos os ficheiros sintetizados, usados para a reconfiguração do TIM. Estes podem ser ficheiros *bitstream* (\*.bit), que definem o código binário usado na reconfiguração do TIM, ou ficheiros *Simple Vector Format* (\*.svf), que contêm o mesmo código binário e um conjunto de instruções para controlar a interface do JTAG usada na ligação de reconfiguração [5].

No painel *Reports* são disponibilizados os relatórios (\*.reg) gerados durante a interação do utilizador com a *RecTool*. No nome dos relatórios estão definidos a data e o horário de criação dos ficheiros [5].

**Weblab Reconfiguration Tool**

Select a file to upload: Escolher ficheiro Nenhum ficheiro selecionado Submit

Allowed files:  
 Build configuration: configuration (\*.conf)  
 Build project: teds (\*.teds), map (\*.map), venlog (\*.v/\*.\*.vh); pinout (\*.ucf)  
 Reconfiguration: svf (\*.svf), bit (\*.bit)

Refresh the panel interface

Server State: Server time: Mon Nov 11 23:59:22  
 Server available for synthesizing the project.

Build			Reconfiguration		
file name	select to build	select to delete			
Configuration files			No project available. Please rebuild the project.		
validation_2.conf	<input type="checkbox"/>	<input type="checkbox"/>	If the project were built, press StartSyn button to create the programming files (*.bit) and (*.svf).		
validation_1.conf	<input checked="" type="checkbox"/>	<input type="checkbox"/>	StartSyn		
Project files			StopSyn		
pinout_1.ucf	<input type="checkbox"/>	<input type="checkbox"/>	file name	select to reconf.	select to delete
Input_8bits_TC_Channel.teds	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-05_14:18:30.bit	<input type="checkbox"/>	<input type="checkbox"/>
map_table_val_2.map	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-05_14:42:43.svf	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Meta_TEDS.teds	<input type="checkbox"/>	<input type="checkbox"/>	reconfigure DELreconf		
Output_6bits_tasks.vh	<input type="checkbox"/>	<input type="checkbox"/>	Reports		
Output_6bits_main.v	<input type="checkbox"/>	<input type="checkbox"/>	file name	select to delete	
XdrcName_TEDS_val_1.teds	<input type="checkbox"/>	<input type="checkbox"/>	svf_2013-03-05_15:27:06.rep	<input type="checkbox"/>	
map_table_val_1.map	<input type="checkbox"/>	<input type="checkbox"/>	Bteds_2013-03-05_15:07:30.rep	<input type="checkbox"/>	
Input_8bits_XdrcName_TEDS_Sbit_input.teds	<input type="checkbox"/>	<input type="checkbox"/>	Syn_2013-03-05_15:08:17.rep	<input type="checkbox"/>	
Output_6bits_tasks_changed.vh	<input type="checkbox"/>	<input type="checkbox"/>	Reconf_2013-10-14_15:58:36.rep	<input type="checkbox"/>	
MPPcontroller_ManufacturerDefined.teds	<input type="checkbox"/>	<input type="checkbox"/>	DELreports		
Input_8bits_tasks.vh	<input type="checkbox"/>	<input type="checkbox"/>			
Input_8bits_main.v	<input type="checkbox"/>	<input type="checkbox"/>			
MPPcontroller_TC_Channel.teds	<input type="checkbox"/>	<input type="checkbox"/>			
pinout_2.ucf	<input type="checkbox"/>	<input type="checkbox"/>			
mpp_lv	<input type="checkbox"/>	<input type="checkbox"/>			
mpp_controller_tasks.vh	<input type="checkbox"/>	<input type="checkbox"/>			
build DELbuild					

Figura 3.6: Interface da ferramenta de reconfiguração de laboratórios remotos.

No painel *Build* estão definidos os ficheiros utilizados na construção do projeto de um laboratório remoto, sendo estes divididos em dois grupos [5]:

- Ficheiros de *Configuração*: são ficheiros de texto que contêm as regras para a reconfiguração do projeto de um laboratório remoto. Determinam ainda as configurações necessárias para interligar os diferentes módulos de laboratório à infraestrutura (\*.conf);
- Ficheiros de *Projeto*: são ficheiros que permitem a compilação de um projeto de um laboratório remoto. Deste grupo fazem parte: i) os ficheiros HDL que descrevem cada módulo de laboratório (\*.v), ii) os ficheiros HDL que definem a interligação dos módulos de laboratório com o módulo IEEE 1451 (\*.vh), iii) os ficheiros binários que descrevem os TEDSs usados pelo módulo IEEE 1451 para aceder aos módulos de laboratório (\*.teds), iv) o ficheiro binário que permite o mapeamento dos TEDSs com o módulo IEEE 1451 (\*.map) e, v) o ficheiro que descreve a saída de pinos usada pela FPGA (\*.ucf).

Apesar da organização dos painéis apresentados na interface da *RecTool*, a criação dos ficheiros de *Configuração* que interligam os módulos de laboratório com a infraestrutura laboratorial remota, bem como os TEDSs que descrevem o comportamento dos TCs e consequentemente dos módulos de laboratório a embutir na infraestrutura é demasiado complexa e demorada. A criação dos TEDSs usados pelo módulo IEEE 1451.0, bem como a geração de um ficheiro que permite mapear esses TEDSs com o respetivo módulo têm subjacente um elevado grau de dificuldade. Adicionalmente, ambos devem ser criados com base em estruturas TLV, definidos através de um conjunto de *bits*, conforme descreve a norma IEEE 1451.0. No caso dos TEDSs, é ainda necessário respeitar a sua estrutura única, tendo em atenção os parâmetros obrigatórios e opcionais de cada um. Nos restantes ficheiros existe a dificuldade de associar corretamente cada um dos TEDSs com os respetivos módulos de laboratório. Desta forma, para facilitar a reconfiguração do laboratório remoto foi desenvolvida uma aplicação *Web* para ler e escrever os diferentes ficheiros utilizados pela *RecTool*.

### **3.5 Sumário**

Neste capítulo foi descrita a família de normas IEEE 1451, em particular a norma IEEE 1451.0. Procedeu-se à explicação da sua arquitetura, destacando-se os três principais componentes: o NCAP, o TIM e os TEDSs.

Por se tratar de um tema revelante para esta dissertação, foi apresentado um protótipo de um laboratório remoto reconfigurável desenvolvido com base numa FPGA seguindo as especificações da norma IEEE 1451.0. No entanto, a complexidade subjacente para criar os TEDSs, que definem o comportamento dos módulos de laboratório e da infraestrutura, bem como os ficheiros para embutir os diferentes módulos na infraestrutura originou a necessidade de se implementar uma aplicação *Web*, cuja descrição é efetuada no capítulo seguinte.

## 4 Descrição da aplicação desenvolvida

O protótipo desenvolvido de um laboratório remoto reconfigurável permite a definição e preparação das experiências através da *RecTool* de forma semelhante à efetuada num laboratório tradicional. Este protótipo foi desenvolvido com base numa FPGA e seguiu as especificações definidas na norma IEEE 1451.0. A definição do comportamento dos módulos de laboratório e da infraestrutura do laboratório remoto obriga à criação de TEDSs, bem como de outros ficheiros para embutir os diferentes módulos de laboratório na infraestrutura. No entanto, devido à complexidade subjacente na preparação de ambos os ficheiros, foi necessário o desenvolvimento de uma aplicação *Web* para simplificar a sua criação.

Neste capítulo é explicada a arquitetura da aplicação desenvolvida, bem como as suas principais funcionalidades. Conclui-se o capítulo descrevendo-se os aspetos de desenvolvimento necessários para implementar a aplicação *Web*, tendo em consideração o *software* escolhido e a base de dados criada.

### 4.1 Motivação subjacente ao desenvolvimento

A arquitetura do protótipo desenvolvido de um laboratório remoto reconfigurável é composta por três partes: i) atores, ii) servidor e, iii) infraestrutura laboratorial remota. A interação com o servidor do laboratório remoto é feita através da *RecTool*. Esta ferramenta permite a reconfiguração da infraestrutura do laboratório com diferentes módulos de laboratório, sendo acedida através da interface apresentada no capítulo anterior, secção 3.4 [5]. Contudo, a reconfiguração de um laboratório remoto através da *RecTool* implica a criação de ficheiros de *Configuração* e de *Projeto*. Ambos têm uma estrutura variável e pressupõem o conhecimento do utilizador para a sua definição.

A criação de um ficheiro de *Configuração*, em que é necessário a interligação dos vários módulos de laboratório com a infraestrutura, sem o apoio de uma aplicação gráfica para distinguir cada um dos módulos, bem como para facilitar a definição das suas relações é exigente para qualquer utilizador. A criação dos ficheiros de *Projeto*, e em particular dos

ficheiros que descrevem os TEDSs, bem como dos utilizados para mapear com o módulo IEEE 1451 (ficheiro *Map Table*), exigem ao utilizador a realização de vários cálculos e conversões.

Deduz-se assim, que as dificuldades encontradas pelos utilizadores no protótipo desenvolvido são originadas pela complexidade na criação dos ficheiros utilizados, devido aos seguintes pontos:

1. Cada TEDSs tem uma estrutura única, em que o preenchimento de cada campo pode variar consoante a sua obrigatoriedade, tipo de dados, limites e ainda a possibilidade de existir dependências com outros campos;
2. Cada valor inserido tem de respeitar a estrutura de dados TLV definida na norma IEEE 1451.0;
3. Dificuldade no cálculo do *checksum* de cada TEDS, de forma a verificar e garantir a integridade dos dados.

Desta forma, é possível compreender a dificuldade subjacente para a criação manual dos ficheiros de *Configuração* e de *Projeto*. Existe uma elevada probabilidade de erro durante a execução de cada um dos passos enumerados, bem como do resultado final dos ficheiros de *Projeto* não estar de acordo com a norma IEEE 1451.0.

Os ficheiros de *Configuração* e de *Map Table* foram criados durante o desenvolvimento do protótipo existente de um laboratório remoto e, por essa razão, não existem ferramentas para a sua definição. Por outro lado, os TEDSs são definidos pela norma IEEE 1451.0 que, apesar de ser uma norma relativamente recente, foi desenvolvida com o objetivo de uniformizar a família de norma IEEE 1451.x. Nesse sentido, foi efetuado um breve levantamento das aplicações existentes relacionadas com a definição de TEDSs segundo a norma IEEE 1451.0. Foram estudadas duas aplicações (*TEDS Compiler* e *TEDS Generator*), cujo resumo é apresentado de seguida.

A aplicação *TEDS Compiler* foi criada após a definição da norma IEEE 1451.0. Esta aplicação foi desenvolvida, no âmbito de uma tese de doutoramento [28], devido à complexidade desta norma na definição dos diferentes TEDSs. Tinha o objetivo de garantir a conformidade destes ficheiros, após a sua criação. Assim, esta aplicação permite invocar através do NCAP vários comandos para o TIM, com o objetivo de verificar a conformidade dos TEDSs com base nas respostas obtidas. No entanto, esta aplicação apresenta várias limitações:

- A verificação do TEDS é efetuada apenas após a sua criação;
- Não disponibiliza as ferramentas necessárias para a leitura de um TEDS;
- Dispõe de pouca informação sobre os campos dos diferentes TEDSs;
- Obriga a que o utilizador conheça a estrutura dos comandos enviados pelo NCAP;
- Não está disponível para *download*.



A aplicação *TEDS Generator* permite a definição de TEDSs com base na norma IEEE 1451.0 [27]. Esta aplicação foi implementada para ser executada localmente com capacidade para criar os seguintes TEDSs: *Meta-TEDS*, *TransducerChannel TEDS*, *End User Application Specific TEDS*, *User's Transducer Name TEDS*, entre outros opcionais. Durante o estudo desta aplicação foram encontradas as seguintes limitações:

- Não permite a leitura de um TEDS já existente;
- Não permite a definição do *Manufacturer-defined TEDS*;
- Não permite o acesso direto a todos os campos numa única página;
- Não descreve os campos dos diferentes TEDSs;
- Execução apenas local;
- Não está disponível para *download*.

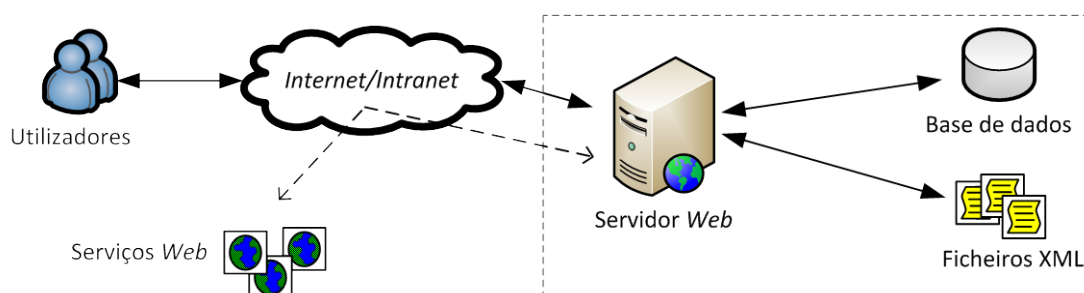
Tendo em consideração as dificuldades encontradas pelos utilizadores e as restrições apresentadas pelas aplicações estudadas, foi desenvolvida uma aplicação *Web* para facilitar a leitura e escrita dos TEDSs, bem como do ficheiro de *Configuração* e *Map Table*. Esta tem como objetivo a disponibilização de várias ferramentas para garantir o apoio à configuração de laboratórios remotos baseados na norma IEEE 1451.0. A utilidade da aplicação surge em dois momentos: i) na definição dos TEDSs e, ii) na definição dos ficheiros que fazem a interligação dos módulos de laboratório com a infraestrutura laboratorial remota. Para uma melhor compreensão do seu funcionamento importa, por isso, apresentar a sua arquitetura e descrever alguns aspetos subjacentes ao seu desenvolvimento.

## 4.2 Arquitetura

Inicialmente, a arquitetura da aplicação desenvolvida tinha como objetivo a sua execução local, pelo que seria descarregada, instalada e utilizada em qualquer máquina compatível. Com o decorrer do projeto o seu objetivo foi alterado para ser uma aplicação *Web*, para que fosse acedida a partir de qualquer lugar, sem a necessidade de descarregar *software* e sem a preocupação da incompatibilidade de versões dos sistemas operativos e dos ambientes de execução.

A aplicação devia assim privilegiar a interatividade com os utilizadores de forma a conseguir recolher toda a informação essencial para os ajudar a realizar a leitura e escrita dos ficheiros. Nesse contexto, a aplicação permite a definição dos ficheiros binários ou de texto em conformidade com a norma IEEE 1451.0, tirando partido das vantagens que uma aplicação *Web* pode garantir, tais como a disponibilidade, a facilidade de comunicação com outros serviços na *Internet* e a interoperabilidade entre plataformas e *Web browsers*.

Conforme representado na figura 4.1, a arquitetura da aplicação é composta por quatro partes: i) utilizadores, atores que interagem com a aplicação, ii) servidor *Web*, que fornece a interface da aplicação permitindo a leitura e escrita dos ficheiros usados pela *RecTool*, iii) serviços *Web*, que permitem a recolha de informação disponível na *Internet* e, iv) ficheiros XML e base de dados, onde estão armazenadas as informações carregadas e consultadas pela aplicação.



**Figura 4.1: Arquitetura da aplicação desenvolvida.**

A arquitetura implementada é suportada por um servidor. O servidor é responsável por disponibilizar a aplicação aos utilizadores. Este integra ainda vários serviços *Web* que facilitam o acesso direto dos utilizadores a informação atualizada e segura na *Internet*. Os serviços *Web* são executados pela aplicação via HTTP através de um URL específico. Nesse URL são enviados os argumentos, que o utilizador introduziu na aplicação, para o servidor. Depois de consultada, a informação é retornada via HTTP em formato XML e interpretada pela aplicação.

A informação gerida pela aplicação é guardada em ficheiros XML e por uma base de dados. Devido à sua flexibilidade, os ficheiros XML guardam a informação essencial para validar os diferentes campos de cada ficheiro, bem como os dados que são lidos em maior quantidade ou que podem necessitar de atualizações durante a vida útil da aplicação. O núcleo de informação da aplicação é garantido pela base de dados. Nesta estão armazenados os dados lidos no momento em que cada página da aplicação é carregada [29].

Com base na arquitetura existente, a aplicação desenvolvida está disponível a qualquer hora e em qualquer lugar, através da *Internet/Intranet*, beneficiando ainda os utilizadores de uma interface bem organizada.

### 4.3 Funcionalidades

A aplicação desenvolvida é acedida pelos utilizadores usando um *Web browser*. Esta permite a leitura e escrita dos ficheiros binários ou de texto, com uma determinada estrutura baseada na norma IEEE 1451.0. A recolha da informação através da sua interface é realizada de forma simples e direta para o utilizador. A aplicação fornece a informação sobre cada campo, garantindo ainda a sua validação de acordo com o tipo de dados esperado. No final, quando

os campos estão preenchidos, a aplicação converte os respetivos dados, realizando automaticamente todos os cálculos necessários. A criação dos ficheiros é feita sem recorrer a outras ferramentas externas, bastando para isso que os utilizadores sigam uma determinada sequência de passos.

### 4.3.1 Ferramentas

Para facilitar a leitura e escrita dos TEDSs foram implementadas várias ferramentas de suporte na aplicação [22]:

- *Meta-TEDS*: define um ficheiro *Meta-TEDS*, cujo objetivo é o de armazenar as informações necessárias para aceder a um qualquer *TransducerChannel*;
- *TranducerChannel TEDS*: define um ficheiro *TranducerChannel TEDS*, cujo objetivo é a descrição de informações específicas do transdutor;
- *User's Transducer Name TEDS*: define um ficheiro *User's Transducer Name TEDS*, cujo objetivo é o de armazenar o nome pelo qual o sistema reconhece o transdutor;
- *End User Application Specific TEDS*: define um ficheiro *End User Application Specific TEDS*, cujo objetivo é o de armazenar um valor variável, em qualquer formato, pelo utilizador;
- *Manufacturer-defined TEDS*: define um ficheiro *Manufacturer-defined TEDS*, cujo objetivo é a construção de um TEDS num formato definido pelo utilizador.

Todas estas ferramentas foram desenvolvidas tendo em consideração as várias estruturas de dados, regras e condições definidas na norma IEEE 1451.0. Desta forma, um TEDS criado com base na norma IEEE 1451.0, independentemente da sua origem, é interpretado por estas ferramentas de suporte.

No protótipo desenvolvido no âmbito de um trabalho de doutoramento [1], a interligação dos diferentes módulos de laboratório com a infraestrutura do laboratório é realizada através de dois ficheiros: *Map Table* e *Configuração*. Para facilitar a leitura e escrita destes ficheiros, foram desenvolvidas na aplicação as ferramentas: *Config Map Table* e *Configure Weblab Modules Connections*.

#### **Config Map Table**

A ferramenta *Config Map Table* permite a leitura e escrita do ficheiro que define o mapeamento dos TEDSs com o módulo IEEE 1451. Assim, os TEDSs do TIM e dos módulos de laboratório, interligados ao módulo IEEE 1451 pelos *TransducerChannels*, são associados a uma determinada posição de memória. O *Map Table* é um ficheiro binário que, apesar de não ser definido pela norma IEEE 1451.0, é criado de acordo com a estrutura TLV, como é ilustrado tabela 4.1. Este ficheiro foi criado durante o trabalho de doutoramento do protótipo de um laboratório remoto reconfigurável [1].

**Tabela 4.1: Formato genérico de um *Map Table*.**

	<b>Campo</b>	<b>Tipo de dados</b>	<b>Octetos</b>	
Estrutura de dados	Tamanho do <i>Map Table</i>	UInt32	4	
	Número de identificação-0	UInt16	2	
	Tamanho	UInt8	1	
	<b>TIM</b>	Código de acesso ao TEDS	UInt8	1
		Posição de memória-0	UInt8	1
		Código de acesso ao TEDS	UInt8	1
		Posição de memória-N	UInt8	1
		Número de identificação-1	UInt16	2
	<b>TC1</b>	Tamanho	UInt8	1
		Código de acesso ao TEDS	UInt8	1
		Posição de memória-N+1	UInt8	1
	<b>TCN</b>	-	UInt8	-
		<i>Checksum</i>	UInt16	2

O primeiro campo representa o número de octetos de toda a estrutura do *Map Table*, definido em quatro octetos através de um valor inteiro positivo, ao qual são adicionados dois octetos do campo *checksum*.

O segundo campo contempla uma estrutura de dados com a informação relativa aos TEDSs do TIM e dos *TransducerChannels*. Para cada um destes, é necessário definir o *Tipo*, que corresponde ao campo número de identificação (número incremental a partir de zero), o *Tamanho* e o *Valor*. O *Valor* é definido pelo campo código de acesso do TEDS e pelo campo posição de memória (número incremental a partir de zero). O *Tamanho* representa a soma de todos os octetos do campo *Valor*. O *Tamanho* é um valor variável devido à possibilidade de existir mais do que um TEDS no TIM ou nos *TransducerChannels*.

O último campo é reservado para o *checksum*. Este tem a responsabilidade de verificar e garantir a integridade do *Map Table*. O cálculo do *checksum* é realizado de forma semelhante à de um TEDS, utilizando a mesma fórmula.

A ferramenta é configurada de acordo com as necessidades do utilizador. Obrigatoriamente a ferramenta define um TIM e um TC. Em cada um deles é necessário a definição de um TEDS. No limite é possível definir até 127 TEDSs no TIM ou nos TCs. O campo número de identificação e posição de memória são calculados automaticamente pela ferramenta.

### **Configure Weblab Modules Connections**

A ferramenta *Configure Weblab Modules Connections* permite a leitura e escrita do ficheiro que define as configurações necessárias para interligar os diferentes módulos de laboratório de infraestrutura. É um ficheiro de texto que não é definido pela norma IEEE 1451.0, nem é criado de acordo com a estrutura TLV. Este ficheiro de *Configuração* foi criado juntamente com o protótipo para criar laboratórios remotos reconfiguráveis, com o objetivo de definir as regras de reconfiguração do projeto desse mesmo laboratório.

De forma geral este ficheiro permite:

- Verificar os TEDSs de cada módulo, bem como a configuração do ficheiro *Map Table*;
- Gerar os ficheiros HDL que descrevem cada módulo de laboratório;
- Interligar os TEDSs e o ficheiro *Map Table* com o módulo IEEE 1451.0.

Resumidamente é possível, através desta ferramenta, a visualização das ligações existentes, mais concretamente dos TCs, entre o módulo IEEE 1451.0 e os diferentes módulos de laboratório. Apresenta ainda os TEDSs associados a cada módulo, bem como os seus atributos e estados. Além disso, mostra ainda as ligações dos módulos de laboratório às experiências, tendo em atenção aos tamanhos e direções configuradas. Para além da descrição das ferramentas, é importante apresentar a organização das suas interfaces, com o objetivo de ajudar os utilizadores na criação e leitura dos diferentes ficheiros.

### 4.3.2 Interfaces

A interface da aplicação *Web* foi construída para funcionar com base nas ações dos utilizadores. Conforme é ilustrado pela figura 4.2, a página inicial é constituída por um índice onde são apresentadas as ferramentas disponíveis na aplicação.

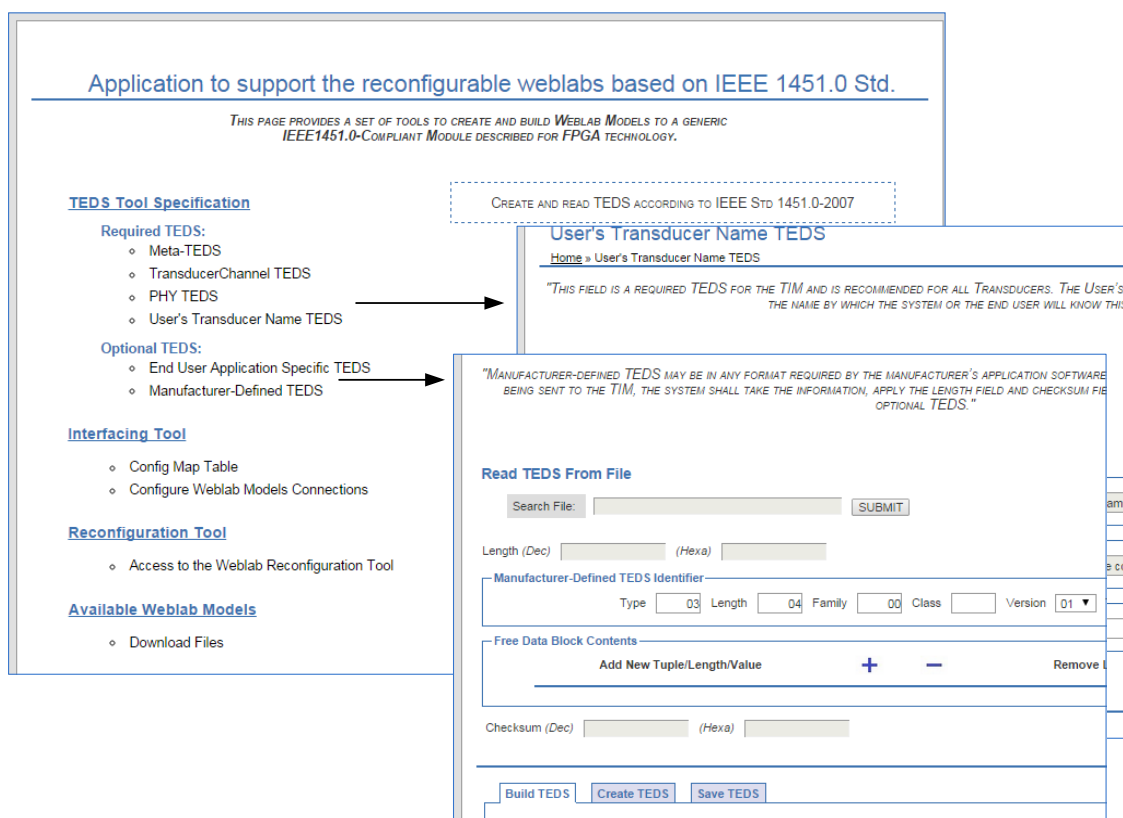


Figura 4.2: Página inicial da aplicação.

Juntamente com estas ferramentas fazem ainda parte deste índice outras páginas de apoio para o funcionamento da aplicação, nomeadamente:

- *TEDS Tool Specification*: apresenta as várias ferramentas para a leitura e escrita dos ficheiros TEDSs obrigatórios ou opcionais (\*.teds);
- *Interfacing Tool*: descreve as ferramentas para a leitura e escrita dos ficheiros que mapeiam os vários módulos de laboratório (\*.map e \*.conf).
- *Reconfiguration Tool*: faz a ligação com a *RecTool*;
- *Available Weblab Modules*: disponibiliza vários módulos de laboratório a embutir na infraestrutura do laboratório remoto.

Com o objetivo de facilitar e enquadrar a navegação do utilizador, em cada uma das opções do índice, é apresentado no lado direito da página inicial uma breve descrição da sua função.

Conforme representado na figura 4.3, a interface<sup>27</sup> de cada ferramenta está dividida em quatro secções principais:

- *Informação*: apresenta o título da página, que corresponde à ferramenta em questão, e uma breve descrição do seu objetivo ou funcionamento. Esta secção apresenta ainda o menu de navegação que permite ao utilizador percorrer as diferentes páginas da aplicação;
- *Upload*: permite ao utilizador importar os diferentes ficheiros permitidos (\*.conf, \*.teds ou \*.map) para serem interpretados e validados pela ferramenta;
- *Formulário*: integra os campos de dados necessários preencher para definir cada um dos ficheiros. Tem uma estrutura dinâmica composta por caixas de texto, seleção e combinação;
- *Download*: contém três painéis distintos: i) *Build TEDS*, é o local onde são feitas as validações de todos os campos da secção do *Formulário* notificando o utilizador quando algum destes não está de acordo com os valores pretendidos, ii) *Create TEDS*, é usado para criar o respetivo ficheiro (binário ou de texto) e apresenta o seu conteúdo de forma estruturada, e iii) *Save TEDS*, é disponibilizado um URL para descarregar o ficheiro criado para o computador do utilizador.

Apesar da organização de cada uma das ferramentas, os utilizadores devem seguir uma sequência de utilização para a leitura e escrita dos diferentes ficheiros.

---

<sup>27</sup> No anexo B são apresentadas as várias interfaces *Web* criadas para cada uma das ferramentas disponíveis na aplicação.

**Figura 4.3:** Interface da ferramenta que permite a escrita, leitura ou alteração do *End User Application Specific TEDS*.

### 4.3.3 Sequência de utilização

A aplicação foi desenvolvida com o intuito do utilizador acompanhar e controlar todas as operações realizadas pelas respetivas ferramentas. A operação mais complexa que um utilizador pode realizar consiste na alteração de um ficheiro já previamente criado. Para isso, o utilizador deve seguir uma sequência de passos que envolve cinco operações principais:

1. O utilizador submete o ficheiro criado na secção de *Upload* da respetiva página da ferramenta. Antes de importar a informação, a ferramenta faz duas validações prévias: i) verifica se o nome do ficheiro tem a extensão prevista (\*.conf, \*.teds ou \*.map) e, ii) verifica a integridade dos dados definidos no ficheiro com base no valor do *checksum*. Esta verificação só é possível porque as ferramentas têm por base a norma IEEE 1451.0. Desta forma, qualquer ficheiro que respeite esta norma é interpretado e importado por esta ferramenta;
2. Depois do ficheiro ser interpretado, o utilizador tem a possibilidade de definir ou alterar qualquer um dos campos na secção do *Formulário*. Os valores predefinidos para cada campo são sugeridos pela ferramenta;
3. Após a alteração dos campos, o utilizador deve garantir a validação dos novos dados. Para isso, o utilizador tem de executar, na secção *Download*, no painel *Build TEDS*, o botão de *Build*. Esta operação verifica para cada campo da secção do *Formulário* se os dados foram ou não introduzidos (se o campo estiver identificado com a expressão

“op.” o seu preenchimento é opcional), se são do tipo esperado (e.g. número inteiro, número decimal, texto, entre outros), ou se o valor está dentro dos limites permitidos. Se o valor introduzido for considerado inválido é apresentada uma mensagem a vermelho no painel *Build* TEDS com a causa do erro. Nesse mesmo instante o campo com o valor errado fica com o fundo a amarelo para facilitar a sua identificação na secção do *Formulário*. Se o valor introduzido for válido, é apresentada uma mensagem de confirmação a preto no painel *Build* TEDS e o fundo do campo verificado fica com a cor branca. Depois de todos os campos serem verificados é apresentada uma mensagem com o estado global da compilação (*BUILD SUCCESSFUL* ou *BUILD FAILED*), como é ilustrado na figura 4.4.

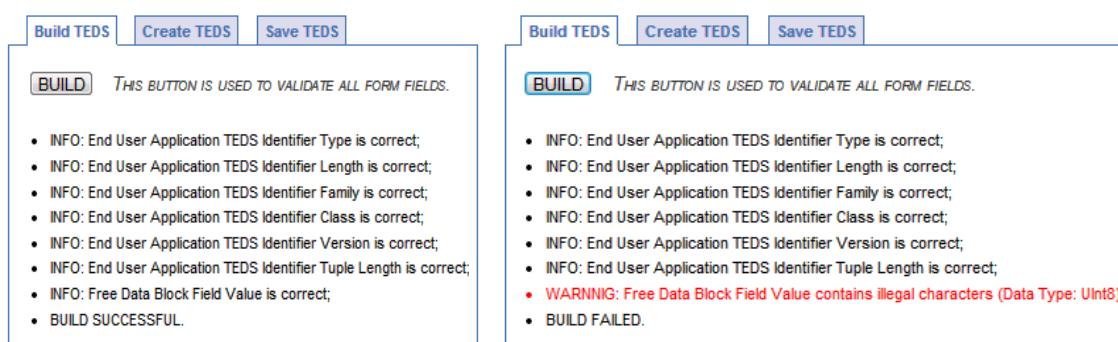


Figura 4.4: Exemplos do registo de atividade do *Build*.

4. Se o estado global da compilação for *BUILD FAILED*, o utilizador deve corrigir todos os erros e executar novamente o botão de *Build* até aparecer a mensagem *BUILD SUCCESSFUL*. Este estado indica que todos os dados introduzidos pelo utilizador foram validados pela ferramenta. Nesse momento, fica disponível para o utilizador o botão de *Create*, disponível na secção *Download*, no painel *Create* TEDS. A operação realizada por este botão serve para criar o ficheiro, seja ele binário ou de texto. Além disso, apresenta ainda os dados do ficheiro de forma estruturada no painel *Create* TEDS, como é ilustrado na figura 4.5.

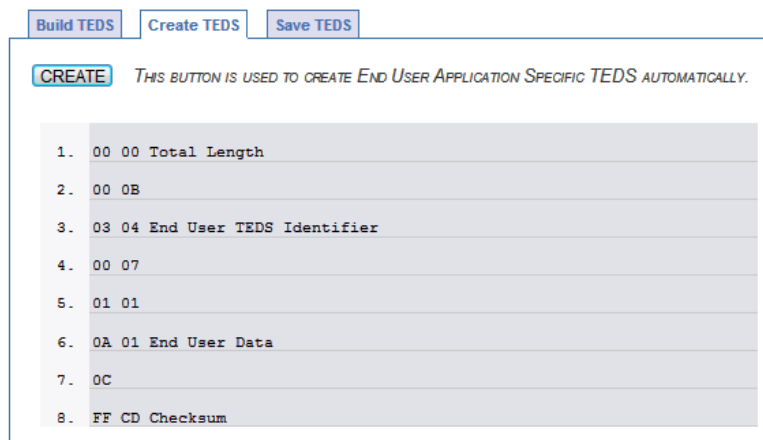


Figura 4.5: Painel com os dados criados do *End User Application Specific* TEDS.



5. Neste passo, o utilizador pode guardar o ficheiro criado. Depois de executar com sucesso o *Build* e de realizar a operação de *Create* fica disponível para o utilizador a hiperligação para descarregar o ficheiro na secção *Download*, no painel *Save TEDS*, como é ilustrado na figura 4.6.



Figura 4.6: Painel com a hiperligação para o ficheiro criado.

A interface da aplicação foi projetada para ser simples, direta e fácil de usar pelos utilizadores. Os aspetos de desenvolvimento necessários para a sua implementação são descritos na subsecção seguinte.

## 4.4 Aspetos de desenvolvimento

Durante a fase de planeamento foi identificado como principal objetivo da aplicação a recolha eficaz dos dados do utilizador, para que estes sejam manuseados e convertidos até serem gerados os ficheiros pretendidos. Desta forma, o *software* adotado deve garantir uma boa usabilidade da aplicação, bem como ter em consideração os meios e o preço necessário para a sua implementação.

### 4.4.1 *Software* utilizado

A aplicação foi desenvolvida num computador pessoal. Para simular o ambiente de um servidor *Web* foi instalado e configurado um servidor Apache<sup>28</sup>, versão 2.2.19 VC9. Este *software* não tem custos associados, é fiável, seguro e pode ser instalado em vários sistemas operativos [30]. Desta forma, através de um URL específico o servidor *Web* disponibiliza a aplicação para os utilizadores.

A aplicação foi desenvolvida e testada para garantir um comportamento idêntico nos seguintes *Web browsers*: i) *Internet Explorer*, superior à versão 10, ii) *Google Chrome*, a partir da versão 7 e, iii) *Mozilla Firefox*, a partir da versão 4.

O SGBD utilizado para funcionar como repositório central da aplicação foi o MySQL<sup>29</sup>, versão 5.5.15. A escolha deste *software* teve em consideração as suas vantagens, tais como: i) rapidez, ii) custos, pois trata-se de um *software* livre, iii) facilidade de utilização, disponibilizando várias ferramentas para ajudar o utilizador na criação de cada base de dados, iv) interoperabilidade, permitindo que seja executada ou instalada em qualquer sistema

<sup>28</sup> <http://www.apache.org/>

<sup>29</sup> <http://www.mysql.com/>

operativo e, v) segurança, garantindo que os utilizadores tenham vários níveis de privilégios no sistema de autorizações, entre outras [29].

A aplicação foi desenvolvida em PHP (*PHP: Hypertext Preprocessor*<sup>30</sup>), versão 5.3.7 VC9. É uma linguagem concebida para ser usada na *Web*. A decisão por esta linguagem foi influenciada pelas suas vantagens, entre as quais se destaca: i) a rapidez, ii) o preço, dado que é um *software* livre, iii) a facilidade de utilização, iv) a interoperabilidade, é compatível com vários sistemas operativos e, v) a segurança, pois o código PHP não é visível para o utilizador [29]. Em combinação com o PHP, foi utilizada a linguagem JavaScript<sup>31</sup> e um conceito denominado AJAX (*Asynchronous JavaScript and XML*<sup>32</sup>), que permitiu a criação de uma aplicação mais dinâmica para o utilizador. Desta forma, é possível obter informações do servidor sem a necessidade de recarregar toda a página da aplicação. Para definir e estruturar o conteúdo de cada página da aplicação foi utilizada a linguagem de estilos CSS (*Cascading Style Sheets*<sup>33</sup>).

Para a criação e edição das imagens utilizadas pela aplicação foi usado o Adobe Photoshop<sup>34</sup>, versão CS4. Foi ainda instalado um *software* de suporte ao desenvolvimento da aplicação denominado phpMyAdmin<sup>35</sup>, versão 3.4.3.2. Este software, desenvolvido em PHP, permite a gestão e monitorização das base de dados do MySQL a partir de um *Web browser*.

Por fim, foi incorporado na aplicação dois serviços *Web* disponibilizados pela *Google Maps API Web Services*<sup>36</sup>: o *Geocoding* e o *Reverse Geocoding*. O *Geocoding* tem como objetivo a conversão de uma morada (e.g. 1600 *Amphitheatre Parkway, Mountain View, CA*) numa coordenada geográfica (e.g. latitude 37.423021 e longitude -122.083739). O *Reverse Geocoding* tem como objetivo a conversão de uma coordenada geográfica num endereço legível [31]. Estes dois serviços são utilizados durante a leitura e escrita do *Meta-TEDS*, convertendo a localização do fabricante do TIM na latitude e longitude correspondente, permitindo igualmente a operação inversa.

#### 4.4.2 Base de dados

A base de dados usada pela aplicação é composta por vinte e sete tabelas<sup>37</sup>, criadas com base na norma IEEE 1451.0 [22]. A construção do modelo relacional da base de dados foi baseada em cinco tabelas principais, cada uma representando um TEDS diferente. A partir destas foram criadas várias tabelas com os dados necessários para criar os TEDSs pelas ferramentas disponíveis na aplicação. Estes dados não devem ser alterados, pois estão definidos de acordo com a norma IEEE 1451.0 [22]. Além disso, a base de dados é apenas utilizada para a consulta

---

<sup>30</sup> <http://php.net/>

<sup>31</sup> <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<sup>32</sup> <https://developer.mozilla.org/pt-PT/docs/AJAX>

<sup>33</sup> <http://www.w3.org/Style/CSS/>

<sup>34</sup> <http://www.adobe.com/pt/products/photoshop.html>

<sup>35</sup> <http://www.phpmyadmin.net>

<sup>36</sup> <https://developers.google.com/maps/documentation/webservices/>

<sup>37</sup> No anexo C é apresentado o modelo relacional da base de dados da aplicação.

de informações, de acordo com as ações dos utilizadores, e para carregar as opções disponíveis nas páginas da aplicação.

Foi ainda necessária a criação de um ficheiro XML de validação para as ferramentas<sup>38</sup> disponíveis na aplicação. Estes ficheiros foram construídos de raiz com base na norma IEEE 1451.0, tendo em consideração as informações dos campos de cada TEDS, nomeadamente: nome, tipo de dados, obrigatoriedade, limite mínimo e máximo, e a mensagem de erro se o campo não for válido durante a execução do *Build*, conforme é representado na figura 4.7.

```
<transducer>
  <teds name="MapTable">
    <field name="MAP" description="Map Table Config">
      <element name="TIM" type="integer" matter="required" min="128" max="255"
        explain=" field has a value between '128' and '255';" />
      <element name="TC" type="integer" matter="required" min="128" max="255"
        explain=" field has a value between '128' and '255';" />
    </field>
  </teds>
</transducer>
```

**Figura 4.7: Exemplo do ficheiro XML para a ferramenta *Config Map Table*.**

Neste momento, foram descritos os principais aspetos de desenvolvimento da aplicação criada para simplificar a definição dos ficheiros de *Configuração* e de *Projeto*. No capítulo seguinte será descrito um cenário prático para realizar a verificação experimental da aplicação desenvolvida.

## 4.5 Sumário

Neste capítulo foi feito um enquadramento com a *RecTool* do protótipo desenvolvido de um laboratório remoto reconfigurável. Salientaram-se as dificuldades atuais na utilização da *RecTool* e as razões que motivaram o desenvolvimento de uma aplicação *Web* para facilitar a leitura e escrita dos ficheiros de *Configuração* e de *Projeto*. De seguida, foi descrita a arquitetura da aplicação, apresentada a interface das várias ferramentas e enumerada a sequência de passos para a sua utilização. Concluiu-se o capítulo com a pormenorização dos aspetos de desenvolvimento da aplicação, cuja verificação experimental é apresentada no capítulo seguinte.

---

<sup>38</sup> No anexo D são detalhados os vários ficheiros XML utilizados para validar os diferentes campos das ferramentas disponíveis na aplicação de suporte.



## 5 Verificação experimental da aplicação

No capítulo anterior foi descrita a aplicação *Web* desenvolvida no âmbito desta dissertação, para simplificar a criação dos módulos de laboratório utilizados na configuração do laboratório. O objetivo deste capítulo consiste em realizar a verificação experimental da aplicação. Este inicia-se com a definição dos objetivos e da estratégia seguida durante esse processo. De seguida, será apresentado um cenário de um laboratório remoto reconfigurável baseado na norma IEEE 1451.0. Com base neste cenário serão criados, a partir da aplicação, os ficheiros TEDSs utilizados para definir os módulos de laboratório, bem como os ficheiros que permitem embutir estes módulos na infraestrutura do laboratório. Por fim, será feita uma avaliação qualitativa da aplicação, referindo-se as suas vantagens e desvantagens em relação à leitura e escrita manual destes ficheiros.

### 5.1 Objetivos e estratégia

A verificação experimental da aplicação segue uma determinada estratégia e compreende três objetivos principais:

1. Descrição de um cenário prático de um laboratório remoto reconfigurável;
2. Criação de ficheiros utilizados na definição de um módulo de laboratório a partir das várias ferramentas disponibilizadas;
3. Avaliação qualitativa da aplicação, comparando os pontos fortes e fracos em relação às dificuldades previamente existentes na criação dos módulos de laboratório utilizados pela *RecTool*, apresentada na subsecção 3.4.3.

Partindo destes objetivos será descrito na subsecção seguinte um cenário prático, que servirá de apoio à verificação experimental da aplicação.

## 5.2 Cenário adotado

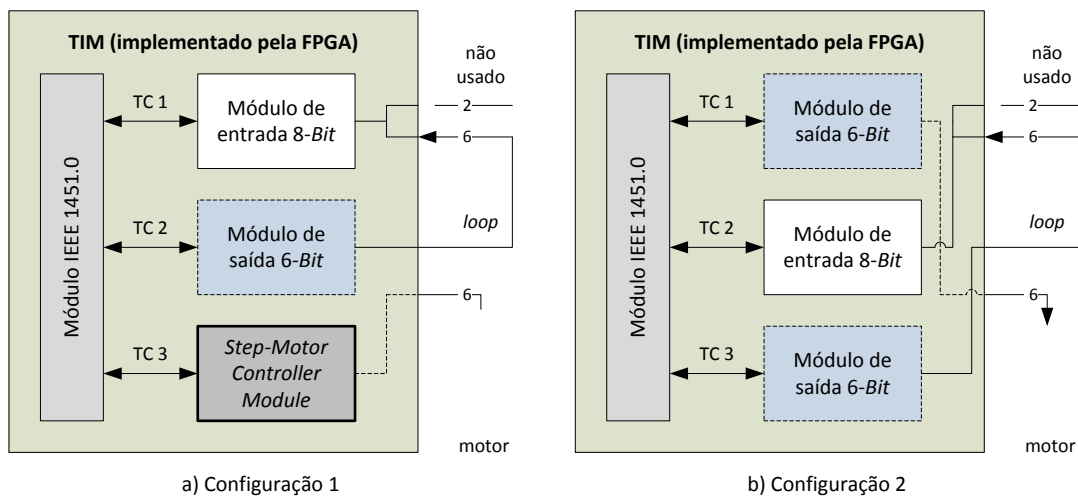
O cenário adotado não tem por objetivo a verificação experimental do protótipo de um laboratório remoto reconfigurável, pois este foi criado, validado e verificado por um trabalho de doutoramento [1][4][32]. Neste sentido, esta secção tem como objetivo a avaliação qualitativa da aplicação *Web* desenvolvida durante a leitura e escrita dos TEDSs utilizados na definição dos módulos de laboratório, bem como dos ficheiros que interligam estes módulos com a infraestrutura do laboratório.

A descrição do cenário adotado obriga à definição de vários módulos de laboratório para serem interligados na infraestrutura do laboratório. Através da *RecTool*, os utilizadores importam os ficheiros que definem cada um dos módulos de laboratório. Em seguida é reconfigurada a infraestrutura, permitindo a troca e configuração desses módulos, sem a necessidade de mudanças físicas na infraestrutura [3][5]. Descreve-se uma possível reconfiguração da infraestrutura com diferentes módulos de laboratório. São descritas duas configurações diferentes utilizando três módulos de laboratório compatíveis com o módulo IEEE 1451.0 predefinido. Estes módulos de laboratório são descritos por ficheiros HDL, de forma a serem utilizados pela *RecTool* na reconfiguração da infraestrutura do laboratório, de acordo com as diferentes configurações. Em cada uma das configurações é possível a execução de duas experiências diferentes, mais concretamente: o controlo de um sinal digital de entrada e saída num circuito de *hardware* e o controlo de um motor passo-a-passo [4][32]. Foram utilizados os seguintes módulos de laboratório:

- Módulo de entrada (*Input*) 8-Bit: monitoriza oito linhas digitais de entrada;
- Módulo de saída (*Output*) 6-Bit: controla seis linhas digitais de saída;
- *Step Motor Controller Module* (SMCM): é um módulo usado para controlar um motor passo-a-passo (velocidade, número de passos, a direção da rotação, entre outros), de acordo com os parâmetros definidos no *Manufacturer-defined* TEDS (MD-TEDS).

A primeira configuração, ilustrada na figura 5.1-a, utiliza três módulos de laboratório. O controlo do circuito de *hardware* é realizado pelos dois módulos digitais de entrada e saída (I/O) usando os TC1 e TC2. O motor passo-a-passo é controlado pelo SMCM acedido através do TC3 [4][32].

A segunda configuração, ilustrada na figura 5.1-b, permite a realização das mesmas experiências referidas na primeira configuração. Nesta configuração, o módulo SMCM é substituído por uma reprodução do módulo de saída de 6-Bit. Os fios externos, que estabelecem a ligação com o circuito de *hardware*, não são alterados. No entanto, os módulos de I/O são reorganizados dentro do TIM, assim como os TCs que permitem o seu controlo. Desta forma, os módulos de I/O adotados para o circuito de *hardware* são controlados pelo TC2 e TC3, respetivamente. O motor passo-a-passo é controlado pelo módulo de saída de 6-Bit acedido através do TC1 [4][32].



**Figura 5.1: Configurações do laboratório remoto.**

(Imagem retirada de R. Costa et al., "A demo prototype of a reconfigurable IEEE 1451.0-compliant and FPGA-based weblabs", 2014)

Após a reconfiguração da infraestrutura com os diferentes módulos de laboratório é possível o controlo destes com a utilização dos comandos definidos pela API HTTP da norma IEEE 1451.0. Estes comandos são enviados para o NCAP, com a informação do TC destino, bem como do TEDS utilizado, entre outros dados. Por fim, é possível aceder e controlar cada um dos módulos de laboratório, assim como as respetivas experiências.

Depois da descrição do cenário adotado, importa referir a importância da aplicação desenvolvida para o apoio na criação dos diferentes módulos de laboratório. Na subsecção seguinte descreve-se a utilização da aplicação para realizar a leitura e escrita dos diferentes ficheiros utilizados na definição dos três módulos de laboratório apresentados.

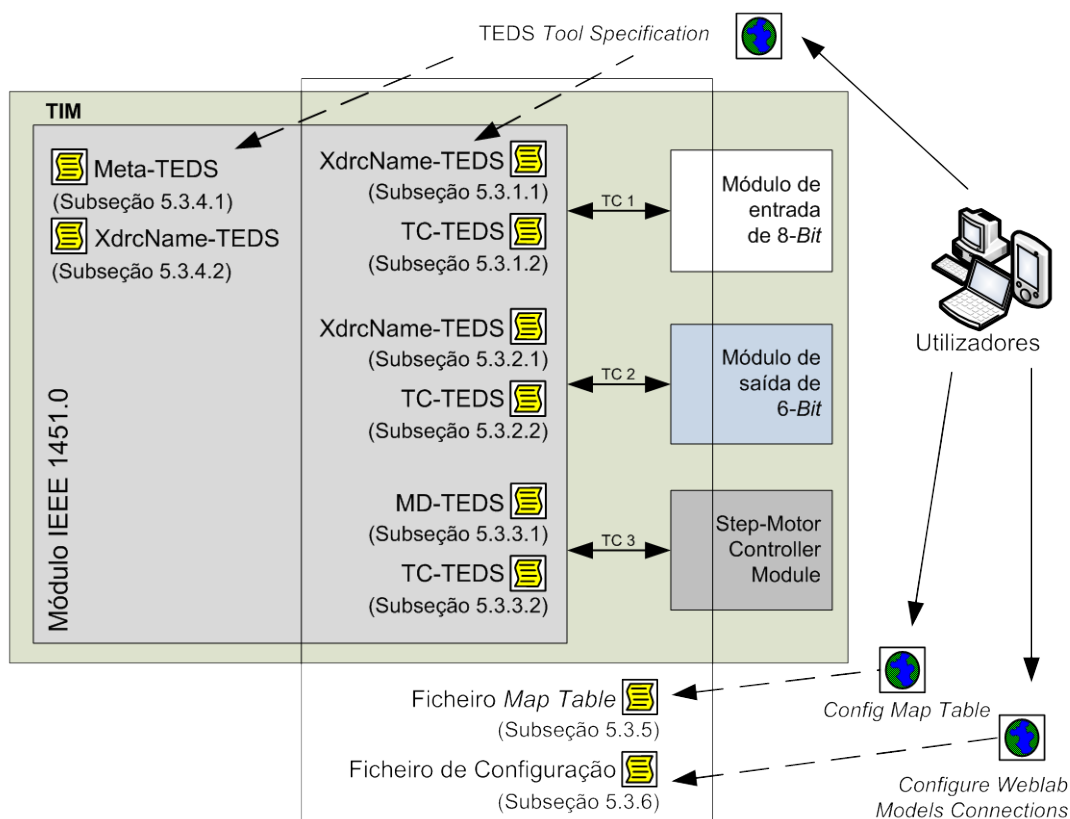
### 5.3 Especificação de ficheiros

As configurações utilizadas obrigam à especificação de vários ficheiros para definir os módulos de laboratório. Para cada módulo são definidos: i) ficheiros HDL (\*.v) para descrever o módulo de laboratório, ii) ficheiros HDL (\*.vh) para interligar estes módulos com o módulo IEEE 1451.0 e, iii) ficheiros TEDSs (\*.teds) que determinam o seu comportamento. No módulo IEEE 1451.0 são também definidos alguns ficheiros TEDSs, com o objetivo de controlar, através dos comandos definidos na norma IEEE 1451.0, os vários módulos de laboratório.

Para facilitar a criação dos diferentes TEDSs, foram desenvolvidas na aplicação várias ferramentas para a sua especificação (*TEDS Tool Specification*), conforme é ilustrado na figura 5.2. Desta forma, os utilizadores ao acederem à aplicação têm acesso a várias ferramentas para a leitura e escrita dos diferentes TEDSs descritos na norma IEEE 1451.0. No cenário adotado os módulos de entrada e de saída utilizam os TEDSs: XdrcName-TEDS e TC-TEDS. Estes são definidos pelas ferramentas *User's Transducer Name TEDS* e *TransducerChannel TEDS* respetivamente. No módulo SMCM são utilizados os TEDSs: MD-TEDS e TC-TEDS. O MD-

TEDS é definido pela ferramenta *Manufacturer-defined TEDS*. No módulo IEEE 1451.0 são utilizados os TEDS: *Meta-TEDS* e *XdrName-TEDS*. O *Meta-TEDS* é definido pela ferramenta com o mesmo nome.

Além dos TEDSs é ainda necessário definir, para as duas configurações apresentadas, o ficheiro *Map Table* e de *Configuração*. O *Map Table* é um ficheiro binário que define o mapeamento dos TEDSs com o módulo IEEE 1451. Este ficheiro é definido pela ferramenta *Config Map Table*. O ficheiro de *Configuração* é um ficheiro de texto que interliga os diferentes módulos de laboratório com a infraestrutura. A sua definição é realizada através da ferramenta *Configure Weblab Modules Connections*.



**Figura 5.2: Diagrama geral da utilização da aplicação desenvolvida.**

Nas subsecções seguintes são apresentados os ficheiros associados a cada módulo do cenário. Nesta apresentação serão definidos os ficheiros TEDSs, o ficheiro *Map Table* e o de *Configuração*, cujas ferramentas foram desenvolvidas durante a realização desta dissertação. Assim, serão apresentados os processos de leitura e escrita destes ficheiros pela aplicação *Web*. Para complementar esta apresentação, foram definidos os seguintes anexos: i) anexo A, que descreve as estruturas de dados dos vários TEDSs definidos na norma IEEE 1451.0, apresentando todos os campos obrigatórios e opcionais e, ii) anexo B, que apresenta as interfaces *Web* das diferentes ferramentas.



### 5.3.1 Módulo de entrada

O módulo de entrada (*Input*) 8-Bit é composto pelos seguintes ficheiros:

- Ficheiro HDL que descreve o módulo de laboratório (*Input\_8bits\_main.v*);
- Ficheiro HDL que define a interligação dos módulos de laboratório com o módulo IEEE 1451 (*Input\_8bits\_tasks.vh*);
- Ficheiros binários que descrevem os TEDSs (*Input\_8bits\_XdrcName.teds* e *Input\_8bits\_TC\_Channel.teds*).

Neste módulo são definidos os ficheiros '*Input\_8bits\_XdrcName.teds*' e '*Input\_8bits\_TC\_Channel.teds*' pelas ferramentas *User's Transducer Name TEDS* e *TransducerChannel TEDS* respetivamente.

#### 5.3.1.1 *User's Transducer Name TEDS*

O ficheiro '*Input\_8bits\_XdrcName.teds*' define um *User's Transducer Name TEDS*, cujo objetivo é o de armazenar o nome pelo qual o sistema reconhece o módulo de entrada. A criação deste TEDS é composta pelas estruturas: i) identificação do TEDS, ii) formato, iii) nome do módulo e, iv) tamanho do TEDS e *checksum* [22].

A identificação do TEDS tem como objetivo a sua descrição. A sua definição é feita com base na norma IEEE 1451.0, como é representada na tabela 5.1. A conversão desta estrutura para o formato TLV origina o resultado: 03.04.00.0C.01.01.

Tabela 5.1: Identificação do *User's Transducer Name TEDS*.

Campo	Valor	Valor (hex)	Descrição
Tipo	03	03	Número do campo.
Tamanho	04	04	Tamanho da estrutura de dados.
Família	00	00	Associado à norma IEEE 1451.0.
Classe	12	0C	Valor que identifica o <i>User's Transducer Name TEDS</i> .
Versão	01	01	Indica que o TEDS está de acordo com a versão inicial da norma IEEE 1451.0.
Tamanho do Tuplo	01	01	Número de octetos do campo tamanho de todos os TLVs.

Na aplicação a identificação do TEDS está definida, por regra, com os mesmos valores apresentados na tabela anterior, como é ilustrado pela figura 5.3. Desta forma, o utilizador não precisa de configurar qualquer valor nesta estrutura de dados. Esta configuração só é possível porque estes valores são constantes na definição deste TEDS.

The image shows a software interface titled "User's Transducer Name TEDS Identifier". It contains several input fields and a dropdown menu. The fields are: "Type" with value 3, "Length" with value 4, "Family" with value 0, "Class" with a dropdown menu showing 12, "XdrcName" with a text input field, "Version" with a dropdown menu showing 01, and "Tuple Length" with value 1.

Figura 5.3: Interface da ferramenta *User's Transducer Name TEDS* – identificação do TEDS.

O formato do TEDS define o conteúdo do bloco de dados. Na aplicação o utilizador define este campo utilizando uma caixa de combinação: '0' se for *Used defined* ou '1' se for *Text-based* TEDS. Ou seja, se o utilizador seleccionar '0' o bloco de dados é definido por ele próprio. Se o utilizador seleccionar '1' então o bloco de dados deve respeitar o formato de um outro TEDS, nomeadamente *Text-based* TEDS. De forma a ajudar o utilizador, é apresentada uma descrição de cada uma das opções, como é ilustrado pela figura 5.4. No cenário adotado foi seleccionada a opção *User-defined*, cujo *Valor* associado é zero. Tendo em consideração o *Tipo* (4) e o *Tamanho* (1), a conversão destes valores para o formato TLV origina o resultado: 04.01.00.



**Figura 5.4: Interface da ferramenta *User's Transducer Name* TEDS – formato do TEDS.**

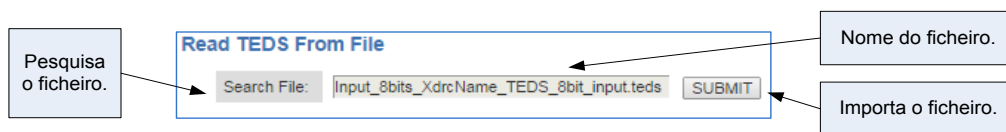
O nome do módulo de entrada é '8-Bit Input'. Na aplicação o utilizador define este campo com base numa caixa de texto. O resultado deste campo é obtido pela conversão direta do nome escolhido em caracteres ASCII. Esta conversão é transparente para o utilizador pois esta é da responsabilidade da aplicação. Tendo em consideração o *Tipo* (5) e o *Tamanho*, que neste caso é variável, a conversão destes valores para o formato TLV origina o resultado: 05.0B.38.2D.42.69.74.20.49.6E.70.75.74.

A criação do *User's Transducer Name* TEDS termina com o cálculo do tamanho do TEDS e do *checksum*. O tamanho total deste TEDS é de 24 octetos e o valor do *checksum* é de 64521. Estes dois campos são calculados e convertidos pela aplicação no momento de criação do TEDS, cujo resultado final é representado na tabela 5.2.

**Tabela 5.2: Formato final do *User's Transducer Name* TEDS.**

ID do campo	Descrição	Tipo de dados	Octetos	Valor (hex)
—	Tamanho do TEDS	UInt32	4	00.00.00.18
0-2	Reservado	-	-	-
3	Identificação do TEDS	UInt8	4	03.04.00.0C.01.01
4	Formato do TEDS	UInt8	1	04.01.00
5	Nome do TIM	-	-	05.0B.38.2D.42.69.74.20.49.6E.70.75.74
—	<i>Checksum</i>	UInt16	2	FC.09

A ferramenta *User's Transducer Name* TEDS permite ainda a leitura de um TEDS deste tipo. Para a leitura do TEDS é necessário que o utilizador selecione o botão *Search file*, que disponibiliza uma janela de pesquisa para escolher o ficheiro pretendido. De seguida, o utilizador executa o botão *Submit* que importa o ficheiro seleccionado na ferramenta *User's Transducer Name* TEDS, como é apresentado na figura 5.5.



**Figura 5.5: Interface da ferramenta *User's Transducer Name* TEDS – leitura do TEDS.**

A importação do ficheiro realiza duas ações principais: i) converte os dados do ficheiro binário para hexadecimal e, ii) valida se o ficheiro selecionado foi criado de acordo com a norma IEEE 1451.0, tendo em consideração o valor do *checksum*, bem como a estrutura TLV dos diferentes tuplos. Se o ficheiro for considerado inválido, é apresentada uma mensagem de erro e os dados não são importados. Se o ficheiro for válido, os dados são importados e associados à ferramenta de acordo com o *Tipo* de cada campo, como é ilustrado na figura 5.6. No entanto, apesar da validade do ficheiro, existe a possibilidade do *Valor* de um dos campos não estar configurado corretamente (e.g. estar definido fora dos limites, estar definido com um diferente tipo de dados, entre outros). Nesse momento, a importação do ficheiro termina e é apresentada no topo da página uma mensagem de erro com o motivo dessa paragem.

**Figura 5.6: Interface da ferramenta *User's Transducer Name* TEDS – bloco de dados.**

A alteração de um valor do *User's Transducer Name* TEDS é um processo igualmente simples. Após a importação do ficheiro, o utilizador tem permissão para alterar qualquer valor definido pelo TEDS, com a exceção dos campos criados automaticamente pela ferramenta (e.g. *checksum*, tamanho do TEDSs, entre outros). A ferramenta *User's Transducer Name* TEDS agrupa os vários campos do TEDS pelo mesmo tipo de informação. Esta apresentação é realizada através de caixas de texto, combinação e seleção, facilitando qualquer possível alteração por parte do utilizador. No final de qualquer alteração é essencial voltar a executar o botão *Build*, para validar os novos dados, e o botão *Create*, para criar novamente o ficheiro.

### 5.3.1.2 *TransducerChannel* TEDS

O ficheiro 'Input\_8bits\_TC\_Channel.teds' define um *TransducerChannel* TEDS, cujo objetivo é a descrição do TC que controla o módulo de laboratório associado. Este TEDS contém informações relacionadas com o tipo de módulo, o modo de funcionamento, o parâmetro físico monitorizado, entre outros [22].

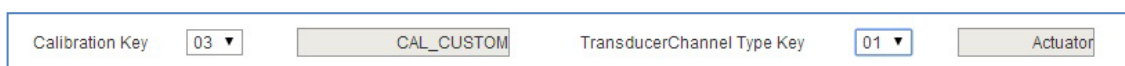
A criação deste TEDS é composta pelas estruturas: i) identificação do TEDS, ii) *calibration key* e *TransducerChannel type key*, iii) *physical units*, iv) *operational lower range limit*, *operational upper range limit* e *worst-case uncertainty*, v) *self-test key*, vi) *sample definition*, vii) *data set*

definition, viii) timing-related information, ix) sampling attribute e, x) tamanho do TEDS e checksum [22].

A identificação do TEDS é feita com base na norma IEEE 1451.0. Na aplicação a identificação do TEDS está definida, por regra, com valores constantes, pois a única diferença entre cada TEDS está no campo classe, que é atribuída automaticamente pela ferramenta. Desta forma, o utilizador não precisa de configurar qualquer valor nesta estrutura de dados. A conversão deste campo para o formato TLV origina o resultado: 03.04.00.03.01.01

O campo *calibration key* define um valor numérico que descreve a calibração do transdutor. No cenário o transdutor não precisa de informação de calibração. Assim, o *Valor* deste campo será zero (CAL\_NONE). Tendo em consideração o *Tipo* (10) e o *Tamanho* (1), a conversão destes valores para o formato TLV origina o seguinte resultado: 0A.01.00. O *TransducerChannel type key* define um valor numérico que descreve o tipo de transdutor (sensor, atuador ou sensor de eventos). No cenário o tipo de transdutor é um sensor. Desta forma, o *Valor* deste campo será zero (sensor). Considerando o *Tipo* (11) e o *Tamanho* (1), a conversão destes valores para o formato TLV origina o resultado: 0B.01.00.

A definição destes campos utilizando a aplicação é feita com base numa secção de combinação, como é ilustrado na figura 5.7. Para o campo *calibration key* foram definidas as combinações: '00-CAL\_NONE', '01-CAL\_SUPPLIED', '03-CAL\_CUSTOM', '04-TIM\_CAL\_SUPPLIED', '05-TIM\_CAL\_SELF' e '06-TIM\_CAL\_CUSTOM'. Para o campo *TransducerChannel type key* foram definidas as combinações: '00-Sensor', '01-Actuador' e '02-Sensor de Eventos'. Com o objetivo de ajudar o utilizador, para ambos os campos, a ferramenta apresenta a respetiva descrição de cada opção.



**Figura 5.7: Interface da ferramenta *TransducerChannel* TEDS – *calibration key* e *TC type key*.**

A estrutura de dados *physical units* define as unidades SI para medir ou controlar uma determinada grandeza física (Radianos, Metros, Quilogramas, Segundos, entre outros). Esta estrutura de dados, cujo *Tipo* é 12, é composta por um tuplo com um ou mais subtuplos. O primeiro subtuplo é representado por um valor numérico que traduz a interpretação das unidades físicas que o módulo de laboratório monitoriza. Os restantes definem um valor para cada unidade SI. A definição desta estrutura de dados é feita com base numa caixa de combinação. O valor numérico definido pela caixa de combinação está associado à forma como são interpretadas as unidades físicas. Se a combinação for: '00-PUI\_SI\_UNITS', '01-PUI\_RATIO\_SI\_UNITS', '02-PUI\_LOG10\_SI\_UNITS' ou '03-PUI\_LOG10\_RATIO\_SI\_UNITS', significa que o transdutor monitoriza uma determinada grandeza física que o utilizador deverá definir. Se a combinação for: '04-PUI\_DIGITAL\_DATA' ou '05-PUI\_ARBITRARY', cada um dos campos fica bloqueado para que não seja possível definir nenhuma grandeza física, como é ilustrado na figura 5.8. No cenário os dados não são representados por unidades SI. Desta forma, o *Valor* da interpretação das unidades físicas será quatro (PUI\_DIGITAL\_DATA). Considerando o *Tipo* (50) e o *Tamanho* (1), a conversão destes valores para o formato TLV

origina o resultado: 32.01.04. Como neste exemplo não é possível definir os valores utilizados em unidades SI, a configuração dos outros subtuplos é opcional. Desta forma, o resultado final da estrutura de dados *physical units* é 0C.03.32.01.04.

**Figura 5.8: Interface da ferramenta *TransducerChannel* TEDS – *physical units*.**

O campo *operational lower range limit* define o valor mínimo que o transdutor vai medir ou controlar. O campo *operational upper range limit* define o valor máximo que o transdutor vai medir ou controlar. O campo *worst-case uncertainty* descreve a variação de erro que ocorre na saída dos dados do tradutor. A definição desta estrutura de dados é feita com base numa caixa de combinação. O valor numérico definido pela caixa de combinação está associado ao valor da estrutura de dados *physical units*. Desta forma, o *Valor* destes campos não é aplicável (NaN = 0x7FFFFFFF) ou zero, como é ilustrado na figura 5.9. Como ambos os campos são obrigatórios para a definição do *TransducerChannel* TEDS, a conversão destes valores no formato TLV é 0D.04.7F.FF.FF.FF, 0E.04.7F.FF.FF.FF e 0F.04.00.00.00.00 respetivamente.

**Figura 5.9: Interface da ferramenta *TransducerChannel* TEDS – *operational lower range limit*, *operational upper range limit* e *worst-case uncertainty*.**

O campo *self-test key* define a capacidade do *TransducerChannel* TEDS fazer testes automáticos. No cenário adotado não são definidos testes automáticos. Na aplicação a definição deste campo é feita com base numa secção de combinação, como é ilustrado na figura 5.10. A configuração deste campo assume dois valores: zero, se não estiverem definidos quaisquer testes automáticos ou um, se forem fornecidas as funções de teste. Considerando o *Tipo* (16) e o *Tamanho* (1), a conversão destes valores para o formato TLV origina o resultado: 10.01.00.

**Figura 5.10: Interface da ferramenta *TransducerChannel* TEDS – *self-test key*.**

A estrutura de dados *sample definition*, cujo *Tipo* é 18, é definida por três subcampos: i) *data model*, que descreve o modelo usado pelos comandos de leitura e escrita do TC para definir um segmento de dados, ii) *data model length*, que define o número de octetos representados pelo subcampo anterior e, iii) *model significant bits*, que configura o número de *bits* significativos do modelo de dados escolhido.

No cenário foi configurado o modelo de dados *Bit sequence*, cujo *Valor* numérico associado é quatro. Considerando o *Tipo* (40) e o *Tamanho* (1) do campo, a conversão destes valores para o formato TLV origina o resultado: 28.01.04. Partindo deste resultado, e tendo em consideração o *Tipo* (41), o *Tamanho* (1) e o *Valor* (1) do campo *data model length*, a conversão destes valores para o formato TLV origina o resultado: 29.01.01. A sequência de *bits* definida tem o *Valor* mil. Tendo em consideração o *Tipo* (42) e o *Tamanho* (2), a conversão destes valores para o formato TLV origina o resultado: 2A.02.00.08. Desta forma, o resultado final desta estrutura é 12.0A.28.01.04.29.01.01.2A.02.00.08.

A definição do campo *data model* é feita com base numa secção de combinação, como é ilustrado na figura 5.11. Para cada valor numérico, a aplicação descreve o respetivo modelo de dados. Desta forma, a partir do valor definido pelo utilizador no campo *model significant bits*, a aplicação determina o *Valor* do *data model length*. Por essa razão, este campo é bloqueado para não ser alterado pelo utilizador.

**Figura 5.11: Interface da ferramenta *TransducerChannel* TEDS – *sample definition*.**

A estrutura de dados *data set definition*, cujo *Tipo* é 19, é definida por cinco subcampos: i) *maximum data repetitions*, que define o número de amostras sobre meio, ii) *series origin*, que descreve uma variável associada com o primeiro segmento de dados, iii) *series increment*, que representa o espaço mínimo entre a variável associada com o primeiro segmento de dados e as sucessivas recolhas de informação, iv) *series units*, que define as unidades físicas dos dois últimos subcampos e, v) *maximum pre-trigger samples*, que representa o número de recolhas realizadas antes de um determinado evento.

No cenário o número máximo de repetições definido é de um. Considerando o *Tipo* (43) e o *Tamanho* (2), a conversão destes valores para o formato TLV origina o resultado: 2B.02.00.01. Como este *Valor* é superior a zero, a definição desta estrutura é obrigatória, como é ilustrado na figura 5.12. Se o *Valor* deste campo for zero, todos os outros campos são bloqueados e ignorados. Neste cenário, os campos *series origin*, *series increment* e *maximum pre-trigger* foram definidos com *Valor* zero. O campo *series units* é definido com base na estrutura de dados *physical units*.

**Figura 5.12: Interface da ferramenta *TransducerChannel* TEDS – *data set definition*.**

A estrutura de dados *timing-related information* é definida por sete campos: i) *update time*, que define o tempo máximo entre a execução do evento e a recolha do primeiro segmento de dados, ii) *write setup time*, que representa o tempo mínimo entre a escrita de um segmento

de dados e a execução de um novo evento, iii) *read setup time*, que configura o tempo máximo entre a receção do evento pelo TIM e o tempo que a informação fica disponível para leitura, iv) *sampling period*, que descreve o período de tempo mínimo para recolha da informação, v) *warm-up time*, que define o período de tempo para que o TC estabilize, vi) *read delay time*, que contém o tempo máximo entre a receção de um comando de leitura de um segmento de dados e o início da transmissão dessa informação e, vii) *self-test requirement*, que representa o tempo máximo necessário para executar um teste automático.

No cenário os intervalos de tempo foram definidos com o *Valor* zero. Na aplicação estes campos são definidos com recurso a uma caixa de texto. Apesar do tipo de transdutor influenciar a obrigatoriedade de alguns campos, estes continuam a ser passíveis de definição, como é ilustrado na figura 5.13.

**Figura 5.13: Interface da ferramenta *TransducerChannel* TEDS – *timing-related information*.**

A estrutura de dados *sampling attribute* é definida por dois subcampos: *sampling mode capability* e *default sampling mode*. O campo *sampling mode capability* define quais os modos de recolha de dados que o *TransducerChannel* TEDS permite. Na aplicação a definição deste campo é feita com base numa caixa de combinação. O valor numérico definido pela caixa de combinação está associado ao modo do funcionamento do transdutor: ‘00–Trigger Initiated Mode’, ‘01–Free-running Without Pre-trigger Mode’, ‘02–Free-running With Pre-trigger Mode’, ‘03–Continuous Sampling Mode’ e ‘04–Immediate Operation Sampling Mode’. Desta forma, para cada opção selecionada pelo utilizador, é apresentada uma mensagem correspondente ao modo de funcionamento do transdutor, como é ilustrado na figura 5.14. No cenário adotado, o transdutor funciona em *Free-running Without Pre-trigger Mode*. Tendo em consideração o *Tipo* (48) e o *Tamanho* (1), a conversão destes valores para o formato TLV origina o resultado: 30.01.01.

O campo *default sampling mode* define o modo de recolha de dados do TC. Neste caso, como o valor é igual ao campo anterior, esta definição é ignorada. Desta forma, o resultado final do campo *sampling attribute* é: 1F.03.30.01.01.

**Figura 5.14: Interface da ferramenta *TransducerChannel* TEDS – *sampling attribute*.**

A criação do *TransducerChannel* TEDS termina com o cálculo do tamanho do TEDS e do *checksum*. O tamanho total deste TEDS é de 129 octetos e o valor do *checksum* é de 62611. Estes dois campos são calculados e convertidos pela aplicação no momento de criação do TEDS, cujo resultado final é representado na tabela 5.3.

**Tabela 5.3: Formato final do *TransducerChannel* TEDS.**

ID do campo	Descrição	Tipo de dados	Octetos	Valor (hexa)
—	Tamanho do TEDS	UInt32	4	00.00.00.81
0-2	Reservado	-	-	-
3	Identificação do TEDS	UInt8	4	03.04.00.03.01.01
10	<i>Calibration key</i>	UInt8	1	0A.01.00
11	<i>Type key</i>	UInt8	1	0B.01.00
12	<i>Physical units</i>	UNITS	-	0C.03.32.01.04
13	<i>Operational lower limit</i>	Float32	4	0D.04.7F.FF.FF.FF
14	<i>Operational upper limit</i>	Float32	4	0E.04.7F.FF.FF.FF
15	<i>Worst-case conditions</i>	Float32	4	0F.04.00.00.00.00
16	<i>Self-Test Key</i>	UInt8	1	10.01.00
18	<i>Sample definition</i>	-	-	12.0A
40	<i>Data model</i>	UInt8	1	28.01.04
41	<i>Data model length</i>	UInt8	1	29.01.01
42	<i>Model significant bits</i>	UInt16	2	2A.02.00.08
19	<i>Data set definition</i>	-	-	13.1C
43	<i>Maximum data rep.</i>	UInt16	2	2B.02.00.01
44	<i>Series origin</i>	Float32	4	2C.04.00.00.00.00
45	<i>Series increment</i>	Float32	4	2D.04.00.00.00.00
46	<i>Series units</i>	UNITS	-	2E.06.32.01.00.33.01.00
47	<i>Maximum pre-trigger</i>	UInt16	2	2F.02.00.00
20	<i>Update time</i>	Float32	4	14.04.00.00.00.00
21	<i>Write setup time</i>	Float32	4	15.04.00.00.00.00
22	<i>Read setup time</i>	Float32	4	16.04.00.00.00.00
23	<i>Sampling period</i>	Float32	4	17.04.00.00.00.00
24	<i>Warn-up period</i>	Float32	4	18.04.00.00.00.00
25	<i>Read delay time</i>	Float32	4	19.04.00.00.00.00
26	<i>Self-test time</i>	Float32	4	1A.04.00.00.00.00
31	<i>Sampling attribute</i>	-	-	1F.03
48	<i>Sampling mode cap.</i>	UInt8	1	30.01.01
-	<i>Checksum</i>	UInt16	2	F4.93

A ferramenta *TransducerChannel* TEDS permite ainda a leitura de um ficheiro TEDS deste tipo. Com o objetivo de facilitar a utilização das diferentes ferramentas, a leitura de um TEDS é efetuada seguindo os mesmos princípios. Nesse sentido, o funcionamento desta ferramenta para realizar a leitura de um TEDS segue a mesma sequência de utilização da ferramenta *User's Transducer Name* TEDS, descrita na subsecção 5.3.1.1. Ambos os ficheiros TEDSs do módulo de entrada (*Input*) 8-*Bit* foram definidos. Desta forma, foi concluído o teste relativo à definição dos ficheiros utilizados neste módulo de laboratório.

### 5.3.2 Módulo de saída

O módulo de saída (*Output*) 6-*Bit* é composto pelos seguintes ficheiros:

- Ficheiro HDL que descreve o módulo de laboratório (*Output\_6bits\_main.v*);
- Ficheiros HDL que definem a interligação dos módulos de laboratório com o módulo IEEE 1451 (*Output\_6bits\_tasks.vh* e *Output\_6bits\_tasks\_changed.vh*);



- Ficheiros binários que descrevem os TEDSs (*Output\_6bits\_TC\_Channel.teds*, *Output\_6bits\_XdrcName.teds* e *Output\_6bits\_XdrcName\_step\_motor.teds*).

Neste módulo são definidos os ficheiros '*Output\_6bits\_XdrcName.teds*' e '*Output\_6bits\_XdrcName\_step\_motor.teds*' pela ferramenta *User's Transducer Name TEDS*, e o ficheiro '*Output\_6bits\_TC\_Channel.teds*' pela ferramenta *TransducerChannel TEDS*.

### 5.3.2.1 *User's Transducer Name TEDS*

Os ficheiros '*Output\_6bits\_XdrcName.teds*' e '*Output\_6bits\_XdrcName\_step\_motor.teds*' definem um *User's Transducer Name TEDS*, cujo objetivo é o de armazenar o nome pelo qual o sistema reconhece o módulo de saída. Como no módulo de entrada foi definido um TEDS do mesmo tipo, nesta subsecção apenas será feita uma breve descrição da estrutura de dados destes ficheiros.

A identificação do TEDS tem como objetivo a sua descrição. A aplicação define, por regra, os valores corretos para a identificação do TEDS. O formato do TEDS define o conteúdo do bloco de dados. Foi definida a opção *User-defined*. O nome do módulo de saída é "6-Bit Output". Na aplicação o utilizador define este campo através de uma caixa de texto. A criação do *User's Transducer Name TEDS* termina com o cálculo do tamanho do TEDS e do *checksum*. O tamanho total deste TEDS é de 25 octetos e o valor do *checksum* é de 64392. Estes dois resultados são calculados e convertidos pela aplicação no momento de criação do TEDS.

O ficheiro '*Output\_6bits\_XdrcName\_step\_motor.teds*' define um *User's Transducer Name TEDS*. Este TEDS difere do anterior apenas na definição do nome do módulo, cuja descrição é "6-Bit Output step-motor control". A criação do *User's Transducer Name TEDS* termina com o cálculo do tamanho do TEDS e do *checksum*. O tamanho total deste TEDS é de 44 octetos e o valor do *checksum* é de 62471. Estes dois resultados são calculados e convertidos pela aplicação no momento de criação do TEDS.

A ferramenta *User's Transducer Name TEDS* permite ainda a leitura de um ficheiro TEDS deste tipo, assim como foi descrito na subsecção 5.3.1.1.

### 5.3.2.2 *TransducerChannel TEDS*

O ficheiro '*Output\_6bits\_TC\_Channel.teds*' define um *TransducerChannel TEDS*, cujo objetivo é a descrição do TC que controla o módulo de laboratório. Como no módulo de entrada foi definido um TEDS do mesmo tipo, nesta subsecção apenas será feita uma breve descrição da estrutura de dados do ficheiro.

A identificação do TEDS tem como objetivo a sua descrição. A aplicação define, por regra, os valores corretos para a identificação do TEDS.

O campo *calibration key* define um valor numérico que descreve a calibração do transdutor. Neste cenário, o transdutor não precisa de informação de calibração. Assim, o *Valor* deste campo será zero (*CAL\_NONE*). O *TransducerChannel type key* define um valor numérico que

descreve o tipo de transdutor (sensor, atuador ou sensor de eventos). O tipo de transdutor é um atuador. Na aplicação a definição destes campos é feita com base numa secção de combinação.

A estrutura de dados *physical units* define as unidades SI para medir ou controlar uma determinada grandeza física. Esta estrutura de dados é composta por um tuplo com um ou mais subtuplos. O primeiro subtuplo é representado por um valor numérico que traduz a interpretação das unidades físicas que o módulo de laboratório deve monitorizar. Os outros subtuplos definem um valor para cada unidade SI. Neste cenário, são utilizados dados digitais que não são representados por unidades SI.

O campo *operational lower range limit* define o valor mínimo que o transdutor vai medir ou controlar. O campo *operational upper range limit* define o valor máximo que o transdutor vai medir ou controlar. O campo *worst-case uncertainty* descreve a variação de erro que ocorre na saída dos dados do tradutor. A definição desta estrutura de dados é feita com base numa caixa de combinação. O valor numérico definido pela caixa de combinação está associado ao valor da estrutura de dados *physical units*. Assim, o Valor destes campos não é aplicável.

O campo *self-test key* define a capacidade do *TransducerChannel* TEDS fazer testes automáticos. Neste cenário não são definidos testes automáticos. A definição deste campo é feita com base numa secção de combinação.

A estrutura de dados *sample definition* é definida por três subcampos: i) *data model*, que descreve o modelo usado pelos comandos de leitura e escrita do TC para definir um segmento de dados, ii) *data model length*, que define o número de octetos representados pelo subcampo anterior e, iii) *model significant bits*, que configura o número de *bits* significativos do modelo de dados escolhido. No cenário foi configurado o modelo de dados *Bit sequence*, cujo Valor numérico associado é quatro. A definição do campo *data model* é feita com base numa secção de combinação. Para cada valor numérico, a aplicação descreve o respetivo modelo de dados. Desta forma, a partir do valor definido pelo utilizador no campo *model significant bits*, a aplicação determina o Valor do *data model length*. Por essa razão, este campo é bloqueado para não ser alterado pelo utilizador.

A estrutura de dados *data set definition* é definida por cinco subcampos: i) *maximum data repetitions*, que define o número de amostras sobre meio, ii) *series origin*, que descreve uma variável associada com o primeiro segmento de dados, iii) *series increment*, que representa o espaço mínimo entre a variável associada com o primeiro segmento de dados e as sucessivas recolhas de informação, iv) *series units*, que define as unidades físicas dos dois últimos subcampos e, v) *maximum pre-trigger samples*, que representa o número de recolhas realizadas antes de um determinado evento. No cenário o número máximo de repetições definido é de um. Como este Valor é superior a zero, a definição desta estrutura é obrigatória. Na aplicação se o Valor deste campo for zero, todos os outros campos são bloqueados e ignorados. No entanto, se o Valor for superior a zero, os outros campos ficam disponíveis para serem definidos pelo utilizador.

A estrutura de dados *timing-related information* é definida por sete campos: i) *update time*, que define o tempo máximo entre a execução do evento e a recolha do primeiro segmento de dados, ii) *write setup time*, que representa o tempo mínimo entre a escrita de um segmento de dados e a execução de um novo evento, iii) *read setup time*, que configura o tempo máximo entre a receção do evento pelo TIM e o tempo que a informação fica disponível para leitura, iv) *sampling period*, que descreve o período de tempo mínimo para recolha da informação, v) *warm-up time*, que define o período de tempo para que o TC estabilize, vi) *read delay time*, que contém o tempo máximo entre a receção de um comando de leitura de um segmento de dados e o início da transmissão dessa informação e, vii) *self-test requirement*, que representa o tempo máximo necessário para executar um teste automático. No cenário os intervalos de tempo foram definidos com o *Valor* zero. Estes campos são definidos com recurso a uma caixa de texto. Apesar do tipo de transdutor influenciar a obrigatoriedade de alguns campos, estes continuam a ser passíveis de definição.

A estrutura de dados *sampling attribute* é definida por dois subcampos: *sampling mode capability* e *default sampling mode*. O campo *sampling mode capability* define quais os modos de recolha de dados que o *TransducerChannel* TEDS permite. A definição deste campo é feita com base numa caixa de combinação. O valor numérico definido pela caixa de combinação está associado ao modo do funcionamento do transdutor: '00–Trigger Initiated Mode', '01–Free-running Without Pre-trigger Mode', '02–Free-running With Pre-trigger Mode', '03–Continuous Sampling Mode' e '04–Immediate Operation Sampling Mode'. Assim, para cada opção selecionada pelo utilizador, é apresentada uma mensagem correspondente ao modo de funcionamento do transdutor. No cenário o transdutor funciona em *Free-running Without Pre-trigger Mode*. O campo *default sampling mode* define o modo de recolha de dados do TC. Como o *Valor* é igual ao campo anterior, esta definição é ignorada.

O campo *end-of-data-set operation* define as operações que um atuador executa quando atinge o limite de um segmento de dados. No cenário o atuador guarda os últimos dados até receber um novo evento. A definição deste campo é feita com base numa caixa de seleção, cujos valores estão entre zero e sete.

O campo *actuator-halt* define o comportamento do atuador quando este recebe um comando *TransducerChannel Idle*. No cenário o atuador deve manter o estado atual até que retorne ao estado de funcionamento do transdutor. A definição deste campo é feita com base numa caixa de seleção. Esta definição é feita com base numa caixa de seleção, cujos valores estão entre zero e três.

A criação do *TransducerChannel* TEDS termina com o cálculo do tamanho do TEDS e do *checksum*. O tamanho total deste TEDS é de 135 octetos e o valor do *checksum* é de 62534. Estes dois resultados são calculados e convertidos pela aplicação no momento de criação do TEDS. A ferramenta *TransducerChannel* TEDS permite ainda a leitura de um TEDS deste tipo. Com o objetivo de facilitar a utilização das diferentes ferramentas, a leitura de um TEDS é efetuada seguindo os mesmos princípios. Nesse sentido, o funcionamento desta ferramenta para realizar a leitura de um TEDS segue a mesma sequência de utilização da ferramenta *User's Transducer Name* TEDS, descrita na subsecção 5.3.1.1.

Ambos os ficheiros TEDSs do módulo de saída (*Output*) 6-*Bit* foram definidos. Desta forma, foi concluído o teste relativo à definição dos ficheiros utilizados neste módulo de laboratório.

### 5.3.3 Módulo SMCM

O módulo *Step Motor Controller Module* (SMCM) é composto pelos seguintes ficheiros:

- Ficheiros HDL que descrevem o módulo de laboratório (mpp\_1.v, mpp\_2.v, mpp\_button\_controller.v, mpp\_clk\_generator.v e mpp\_controller.v);
- Ficheiro HDL que define a interligação dos módulos de laboratório com o módulo IEEE 1451 (mpp\_controller\_tasks.vh);
- Ficheiros binários que descrevem os TEDSs (MPPcontroller\_ManufacturerDefined.teds e MPPcontroller\_TC\_Channel.teds).

Neste módulo são definidos os ficheiros ‘MPPcontroller\_ManufacturerDefined.teds’ e ‘MPPcontroller\_TC\_Channel.teds’ pela ferramenta *Manufacturer-defined TEDS* e *TransducerChannel TEDS* respetivamente.

#### 5.3.3.1 *Manufacturer-defined TEDS*

O ficheiro ‘MPPcontroller\_ManufacturerDefined.teds’ define um *Manufacturer-defined TEDS*, cujo objetivo é a construção de um TEDS num formato definido pelo utilizador. A criação deste TEDS é composta pelas estruturas: i) identificação do TEDS, ii) bloco de dados e, iii) tamanho do TEDS e *checksum* [22].

A identificação do TEDS tem como objetivo a sua descrição. A aplicação define, por regra, os valores corretos para a identificação do TEDS, com a exceção do campo classe, que determina o código de acesso associado ao tipo deste TEDS. Como é um TEDS definido pelo utilizador, o código de acesso do campo classe assume um valor entre 128 e 255. Neste cenário, o *Valor* é 128. A conversão desta estrutura para o formato TLV origina o resultado: 03.04.00.80.01.01.

O bloco de dados diz respeito a uma estrutura dinâmica definida pelo utilizador. Os campos criados nesta estrutura de dados têm um *Tipo*, *Tamanho* e *Valor* fornecido pelo utilizador, e definidos de acordo com a norma IEEE 1451.0. A aplicação permite ao utilizador a definição de múltiplos campos com vários tipos e formatos, como é ilustrado na figura 5.15. Para isso, o utilizador adiciona um tuplo para criar um novo campo. De seguida, configura o *Tipo* e o respetivo formato (UInt8, UInt16, UInt24, UInt32). Com base no formato, o *Tamanho* (número de octetos) é calculado automaticamente pela aplicação. Neste cenário foi necessário a definição de quatro novos campos: direção, número de passos, modo de funcionamento e velocidade.

Length (Dec)  (Hexa)  Time: 3:21:18 PM Date: 2014-08-29

**Manufacturer-Defined TEDS Identifier**

Type  Length  Family  Class  Version  Tuple Length

**Free Data Block Contents**

Add New Tuple/Length/Value   Remove Last Tuple/Length/Value

Field Type	Data Type	Octects	Value	
<input type="text" value="4"/>	<input type="text" value="UInt8"/>	<input type="text" value="01"/>	<input type="text" value="1"/>	<input type="button" value="X"/>
<input type="text" value="5"/>	<input type="text" value="UInt16"/>	<input type="text" value="02"/>	<input type="text" value="65535"/>	<input type="button" value="X"/>
<input type="text" value="6"/>	<input type="text" value="UInt8"/>	<input type="text" value="01"/>	<input type="text" value="0"/>	<input type="button" value="X"/>
<input type="text" value="7"/>	<input type="text" value="UInt24"/>	<input type="text" value="03"/>	<input type="text" value="100000"/>	<input type="button" value="X"/>

Checksum (Dec)  (Hexa)

**Figura 5.15: Interface da ferramenta *Manufacturer-defined TEDS*.**

O campo direção define o rumo do motor passo-a-passo. Esta definição assume os valores: ‘0-esquerda’ ou ‘1-direita’. No cenário o *Valor* é um. Tendo em consideração o *Tipo* (4) e o *Tamanho* (1), a conversão destes valores para o formato TLV origina o resultado: 04.01.01.

O campo número de passos indica o movimento que o motor passo-a-passo realiza depois de receber um novo evento. No cenário o *Valor* é 65535. Neste caso, o *Valor* é ignorado pelo motor passo-a-passo, que funciona de forma contínua. Tendo em consideração o *Tipo* (5) e o *Tamanho* (2), a conversão destes valores para o formato TLV origina o resultado: 05.02.FF.FF.

O campo modo de funcionamento define a sequência que é possível gerar no motor passo-a-passo pelo utilizador. Esta definição assume os valores: ‘0-meio passo’, ‘1- normal’ ou ‘2-onda’. No cenário o *Valor* é zero. Tendo em consideração o *Tipo* (5) e o *Tamanho* (2), a conversão destes valores para o formato TLV origina o resultado: 06.01.01.

O campo velocidade define o maior sinal *clk* gerado pelo módulo SMCM, de forma a controlar a velocidade gerada pelo motor passo-a-passo. No cenário o *Valor* é 100000. Tendo em consideração o *Tipo* (6) e o *Tamanho* (3), a conversão destes valores para o formato TLV origina o resultado: 07.03.01.86.A0.

A criação do *Manufacturer-defined TEDS* termina com o cálculo do tamanho do TEDS e do *checksum*. O tamanho total deste TEDS é de 23 octetos e o valor do *checksum* é de 64540. Estes dois campos são calculados e convertidos pela aplicação no momento de criação do TEDS, cujo resultado final é representado na tabela 5.4.

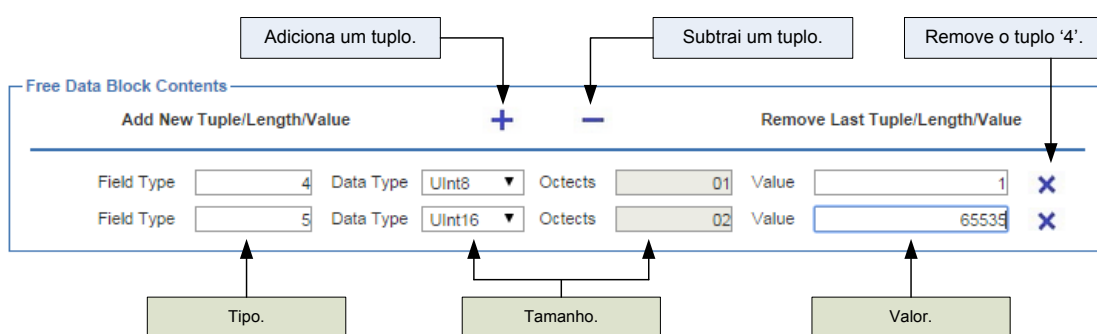
A ferramenta *Manufacturer-defined TEDS* permite ainda a leitura de um TEDS deste tipo. Com o objetivo de facilitar a utilização das diferentes ferramentas, a leitura de um TEDS é efetuada seguindo os mesmos princípios. Nesse sentido, o funcionamento desta ferramenta para realizar a leitura de um TEDS segue a mesma sequência de utilização da ferramenta *User’s Transducer Name TEDS*, descrita na subsecção 5.3.1.1. Adicionalmente, esta ferramenta permite ainda ao utilizador adicionar vários campos, com diferentes tipos de dados. Desta forma, a ferramenta disponibiliza, de forma intuitiva, vários botões para adicionar, subtrair ou

remover novos tuplos na secção *free data block*. Cada tuplo é um novo campo, composto pelo seu *Tipo*, *Tamanho* e *Valor*.

**Tabela 5.4: Formato final do *Manufacturer-defined* TEDS.**

ID do campo	Descrição	Tipo de dados	Octetos	Valor (hex)
—	Tamanho do TEDS	UInt32	4	00.00.00.17
0-2	Reservado	-	-	-
3	Identificação do TEDS	UInt8	4	03.04.00.80.01.01
4	Direção	UInt8	1	04.01.01
5	Número de passos	UInt16	2	05.02.FF.FF
6	Modo funcionamento	UInt8	1	06.01.00
7	Velocidade	UInt24	3	07.03.01.86.A0
—	<i>Checksum</i>	UInt16	2	FC.1C

Conforme é apresentado pela figura 5.16, o utilizador pode adicionar (botão mais), subtrair (botão menos) ou remover (botão cruz) um tuplo diretamente. No final de qualquer alteração é essencial voltar a executar o botão *Build*, para validar os novos dados, e o botão *Create*, para criar novamente o ficheiro.



**Figura 5.16: Interface da ferramenta *Manufacturer-defined* TEDS – *free data block*.**

### 5.3.3.2 *TransducerChannel* TEDS

O ficheiro 'MPPcontroller\_TC\_Channel.teds' define um *TransducerChannel* TEDS, cujo objetivo é a descrição do TC que controla o módulo de laboratório. Como no módulo de entrada foi definido um TEDS do mesmo tipo, nesta subsecção apenas será feita uma breve descrição da estrutura de dados do ficheiro.

A identificação do TEDS tem como objetivo a sua descrição. A aplicação define, por regra, os valores corretos para a Identificação do TEDS.

O campo *calibration key* define um valor numérico que descreve a calibração do transdutor. Neste cenário, o transdutor não precisa de informação de calibração. Assim, o Valor deste campo será zero (CAL\_NONE). O *TransducerChannel type key* define um valor numérico que descreve o tipo de transdutor (sensor, atuador ou sensor de eventos). O tipo de transdutor é um atuador. A definição destes campos é feita com base numa secção de combinação.

A estrutura de dados *physical units* define as unidades SI para medir ou controlar uma determinada grandeza física. Esta estrutura de dados é composta por um tuplo com um ou mais subtuplos. O primeiro subtuplo é representado por um valor numérico que traduz a interpretação das unidades físicas que o módulo de laboratório deve monitorizar. Os outros subtuplos definem um valor para cada unidade SI. Neste cenário, são utilizados dados digitais que não são representados por unidades SI.

O campo *operational lower range limit* define o valor mínimo que o transdutor vai medir ou controlar. O campo *operational upper range limit* define o valor máximo que o transdutor vai medir ou controlar. O campo *worst-case uncertainty* descreve a variação de erro que ocorre na saída dos dados do tradutor. A definição desta estrutura de dados é feita com base numa caixa de combinação. O valor numérico definido pela caixa de combinação está associado ao valor da estrutura de dados *physical units*. Assim, o *Valor* destes campos não é aplicável.

O campo *self-test key* define a capacidade do *TransducerChannel* TEDS fazer testes automáticos. Neste cenário não são definidos testes automáticos. A definição deste campo é feita com base numa secção de combinação.

A estrutura de dados *sample definition* é definida por três subcampos: i) *data model*, que descreve o modelo usado pelos comandos de leitura e escrita do TC para definir um segmento de dados, ii) *data model length*, que define o número de octetos representados pelo subcampo anterior e, iii) *model significant bits*, que configura o número de *bits* significativos do modelo de dados escolhido. No cenário foi configurado o modelo de dados *Bit sequence*, cujo *Valor* numérico associado é seis. A definição do campo *data model* é feita com base numa secção de combinação. Para cada valor numérico, a aplicação descreve o respetivo modelo de dados. Desta forma, a partir do valor definido pelo utilizador no campo *model significant bits*, a aplicação determina o *Valor* do *data model length*. Por essa razão, este campo é bloqueado para não ser alterado pelo utilizador.

A estrutura de dados *data set definition* é definida por cinco subcampos: i) *maximum data repetitions*, que define o número de amostras sobre meio, ii) *series origin*, que descreve uma variável associada com o primeiro segmento de dados, iii) *series increment*, que representa o espaço mínimo entre a variável associada com o primeiro segmento de dados e as sucessivas recolhas de informação, iv) *series units*, que define as unidades físicas dos dois últimos subcampos e, v) *maximum pre-trigger samples*, que representa o número de recolhas realizadas antes de um determinado evento. No cenário o número máximo de repetições definido é de oito. Como este *Valor* é superior a zero, a definição desta estrutura é obrigatória. Se o *Valor* deste campo for zero, todos os outros campos são bloqueados e ignorados. No entanto, se o *Valor* for superior a zero, os outros campos ficam disponíveis para serem definidos pelo utilizador.

A estrutura de dados *timing-related information* é definida por sete campos: i) *update time*, que define o tempo máximo entre a execução do evento e a recolha do primeiro segmento de dados, ii) *write setup time*, que representa o tempo mínimo entre a escrita de um segmento de dados e a execução de um novo evento, iii) *read setup time*, que configura o tempo

máximo entre a receção do evento pelo TIM e o tempo que a informação fica disponível para leitura, iv) *sampling period*, que descreve o período de tempo mínimo para recolha da informação, v) *warm-up time*, que define o período de tempo para que o TC estabilize, vi) *read delay time*, que contém o tempo máximo entre a receção de um comando de leitura de um segmento de dados e o início da transmissão dessa informação e, vii) *self-test requirement*, que representa o tempo máximo necessário para executar um teste automático. No cenário os intervalos de tempo foram definidos com o *Valor* zero. Estes campos são definidos com recurso a uma caixa de texto. Apesar do tipo de transdutor influenciar a obrigatoriedade de alguns campos, estes continuam a ser passíveis de definição.

A estrutura de dados *sampling attribute* é definida por dois subcampos: *sampling mode capability* e *default sampling mode*. O campo *sampling mode capability* define quais os modos de recolha de dados que o *TransducerChannel* TEDS permite. A definição deste campo é feita com base numa caixa de combinação. O valor numérico definido pela caixa de combinação está associado ao modo do funcionamento do transdutor: '00–Trigger Initiated Mode', '01–Free-running Without Pre-trigger Mode', '02–Free-running With Pre-trigger Mode', '03–Continuous Sampling Mode' e '04–Immediate Operation Sampling Mode'. Desta forma, para cada opção seleccionada pelo utilizador, é apresentada uma mensagem correspondente ao modo de funcionamento do transdutor. No cenário o transdutor funciona em *Free-running Without Pre-trigger Mode*. O campo *default sampling mode* define o modo de recolha de dados do TC. Neste caso, como o *Valor* é igual ao campo anterior, esta definição é ignorada.

O campo *end-of-data-set operation* define as operações que um atuador executa quando atinge o limite de um segmento de dados. No cenário o atuador guarda os últimos dados até receber um novo evento. A definição deste campo é feita com base numa caixa de seleção, cujos valores estão entre zero e sete.

O campo *actuator-halt* define o comportamento do atuador quando este recebe um comando *TransducerChannel Idle*. No cenário o atuador deve manter o estado atual até que retorne ao estado de funcionamento do transdutor. A definição deste campo é feita com base numa caixa de seleção. Esta definição deste campo é feita com base numa caixa de seleção, cujos valores estão entre zero e três.

A criação do *TransducerChannel* TEDS termina com o cálculo do tamanho do TEDS e do *checksum*. O tamanho total deste TEDS é de 135 octetos e o valor do *checksum* é de 62534. Estes dois resultados são calculados e convertidos pela aplicação no momento de criação do TEDS. A ferramenta *TransducerChannel* TEDS permite ainda a leitura de um TEDS deste tipo. Com o objetivo de facilitar a utilização das diferentes ferramentas, a leitura de um TEDS é efetuada seguindo os mesmos princípios. Nesse sentido, o funcionamento desta ferramenta para realizar a leitura de um TEDS segue a mesma sequência de utilização da ferramenta *User's Transducer Name* TEDS, descrita na subsecção 5.3.1.1.

Ambos os ficheiros TEDSs do módulo SMCM foram definidos. Desta forma, foi concluído o teste relativo à definição dos ficheiros utilizados neste módulo de laboratório.



### 5.3.4 Módulo IEEE1451.0

O módulo IEEE 1451.0 é composto pelos ficheiros binários que descrevem os TEDSs denominados, 'Meta\_TEDS.teds' e 'XdrcName\_TEDS.teds'. Para a sua definição são utilizadas as ferramentas *Meta-TEDS* e *User's Transducer Name TEDS* respetivamente.

#### 5.3.4.1 Meta-TEDS

O ficheiro 'Meta\_TEDS.teds' define um *Meta-TEDS*, cujo objetivo é o de armazenar as informações necessárias para aceder a um qualquer TC. A criação deste TEDS é composta pelas estruturas: i) identificação do TEDS, ii) UUID, iii) *timing-related information*, iv) *TransducerChannels* e, x) tamanho do TEDS e *checksum* [22].

A Identificação do TEDS tem como objetivo a sua descrição. A sua definição é feita com base na norma IEEE 1451.0. A conversão deste campo para o formato TLV origina o resultado: 03.04.00.01.01.01.

A estrutura de dados UUID define um identificador único universal do TIM. Como é representado na tabela 5.5, a definição deste campo tem em consideração: i) a localização geográfica da construção do TEDS, ii) o código do fabricante, iii) o ano e, iv) a data/sequência. A construção deste campo implica que os dados sejam convertidos para binário e, de seguida, sejam novamente convertidos para hexadecimal. Finalmente são definidos de acordo com a estrutura TLV definida pela norma IEEE 1451.0. Na aplicação a criação deste campo é facilitada. Para a definição da localização geográfica é necessário preencher os campos: endereço, cidade, distrito, código postal e país. Com base nestes campos, a aplicação através de um serviço *Web*, calcula automaticamente as coordenadas geográficas pretendidas, como é ilustrado na figura 5.17. Essas coordenadas são depois apresentadas na interface nos campos latitude e longitude. Estes campos não podem ser alterados diretamente pelo utilizador. Apenas são alterados se algum dos outros campos mencionados for alterado, o que obriga ao cálculo de uma nova coordenada geográfica. A aplicação permite ainda a definição do código do fabricante e da sequência. A definição destes campos é feita através de uma caixa de texto. Por fim, a aplicação disponibiliza um calendário para o utilizador definir o dia de construção do TEDS.

Tabela 5.5: Estrutura de dados UUID.

Campo	Descrição	Valor	Valor (bits)
N/S	Pontos cardeais Norte/Sul	Norte = 1 /Sul = 0	1
Latitude	-	-	20
E/W	Pontos cardeais Este/Oeste	Este = 1 /Oeste = 0	1
Longitude	-	-	20
Código do fabricante	-	0-15	4
Ano	Ano corrente	-	12
Data/sequência	Dia do ano	0-365	9
	Hora	0-24	5
	Sequência	0-256	8

Universal Unique Identification Data

Address  County  PostCode  Manufacturer

Town/City  Latitude  Longitude  Date (yyyy-mm-dd)

Country  Sequence

(Note: Latitude and Longitude are calculated automatically. Please complete all fields.)

**Figura 5.17: Interface da ferramenta Meta- TEDS – UUID.**

No cenário o TIM foi produzido a 11 de Outubro de 2010, às 16 horas. Nesse dia, foi o primeiro TIM a ser produzido pelo fabricante, cujo código de acesso é um. O TIM foi produzido na morada: Rua da Alemã, Vila Nova de Gaia, código-postal 4430, Portugal. As coordenadas geográficas deste local são: N41.1029573 e W-8.5985171. De seguida, todos estes dados são convertidos para binário, com a exceção da latitude e longitude que são convertidos primeiro para arco-segundo (arcseg), conforme é representado na tabela 5.6.

**Tabela 5.6: Conversão para binário da estrutura de dados UUID.**

Campo	Valor	Valor (binário)	Valor (bits)
N/S	N	1	1
Latitude	147970	00100100001000000010	20
E/W	W	0	1
Longitude	30954	11111000011100010110	20
Código do fabricante	1	0001	4
Ano	2010	011111011010	12
Data/sequência	2010-10-11	100011100	9
	16 Horas	10000	5
	1	00000001	8

Por fim, é necessário agrupar os valores em binário por octetos, conforme é representado na tabela 5.7. Tendo em consideração o *Tipo* (4) e o *Tamanho* (10), a conversão destes valores para o formato TLV origina o resultado: 04.0A.92.10.13.E1.C5.85.F6.A3.90.01. Como a realização destas conversões é responsabilidade da aplicação, o risco de erro é reduzido.

**Tabela 5.7: Conversão para hexadecimal da estrutura de dados UUID.**

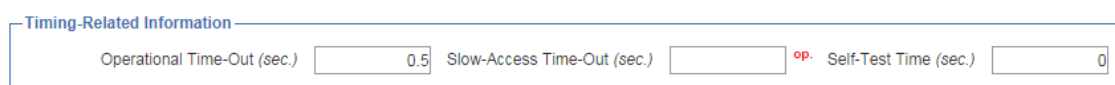
Valor (binário)	Valor (hexa)
10010010	92
00010000	10
00010011	13
11100001	E1
11000101	C5
10000101	85
11110110	F6
10100011	A3
10010000	90
00000001	01

A estrutura de dados *timing-related information* é composta por três campos: *operational time-out*, *slow-access time-out* e *self-test time*. Todos os campos são representados em vírgula flutuante de acordo com a norma IEEE 754<sup>39</sup>.

O campo *slow-access time-out* define o intervalo de tempo, em segundos, para os comandos enviados pelo NCAP que precisem de um maior tempo de execução. É um campo opcional e não é definido neste cenário.

O campo *self-test time* define o maior tempo, em segundos, que o TIM tem para executar um teste automático. É um campo obrigatório. Neste cenário foi definido um período de tempo de zero segundos. Tendo em consideração o *Tipo* (12) e o *Tamanho* (4), a conversão destes valores para o formato TLV origina o resultado: 0C.04.00.00.00.00.

A definição destes campos é feita através de uma caixa de texto, como é ilustrado na figura 5.18. O campo *slow-access time-out* é ainda representado como sendo um campo opcional, através do marcador 'op.' colocado à direita da respetiva caixa de texto.



**Figura 5.18:** Interface da ferramenta *Meta-TEDS – timing-related information*.

A estrutura de dados *TransducerChannels* define o número de TCs implementados no TIM. Este campo é definido por um número incremental a partir de um.

Nesta estrutura de dados é ainda possível associar os TCs a um determinado grupo:

- *Control group*: identifica os transdutores que são usados para controlar as operações de outros transdutores;
- *Vector group*: identifica as interligações das sequências de dados dentro do TIM;
- *Geographic location group*: identifica os TIMs que implementam a informação de localização dinâmica;
- *TransducerChannel proxies group*: combina a informação de saída de vários sensores ou a informação de entrada de vários atuadores numa única estrutura.

No cenário são definidos três TCs. Nenhum destes está associado a um qualquer grupo. Tendo em consideração o *Tipo* (13) e o *Tamanho* (2), a conversão destes valores para o formato TLV origina o resultado: 0D.02.00.03. O número de TCs é definido através de uma caixa de texto. Se o utilizador pretender associar algum dos respetivos TCs a um determinado grupo, apenas tem de seleccionar o botão “*Define Channels*” e realizar a respetiva ação.

<sup>39</sup> A norma IEEE 754-2008 define as regras de normalização para as operações e representações de números binários com vírgula flutuante, de precisão simples (32 bits) e de precisão dupla (64 bits).

A criação do *Meta*-TEDS termina com o cálculo do tamanho do TEDS e do *checksum*. O tamanho total deste TEDS é de 24 octetos e o valor do *checksum* é de 64091. Estes dois campos são calculados e convertidos pela aplicação no momento de criação do TEDS, cujo resultado final é representado na tabela 5.8.

**Tabela 5.8: Formato final do *Meta*-TEDS.**

ID do campo	Descrição	Tipo de dados	Octetos	Valor (hexa)
—	Tamanho do TEDS	UInt32	4	00.00.00.24
0-2	Reservado	-	-	-
3	Identificação do TEDS	UInt8	4	03.04.00.01.01.01
4	UUID	-	10	04.0A.92.10.13.E1.C5.85.F6.A3.90.01
10	<i>Operational time-out</i>	Float32	4	0A.04.3F.00.00.00.00
12	<i>Self-time time</i>	Float32	4	0C.04.00.00.00.00
13	Número de TCs	UInt16	2	0D.02.00.03
-	<i>Checksum</i>	UInt16	2	FA.5B

A ferramenta *Meta*-TEDS permite ainda a leitura de um ficheiro TEDS deste tipo. Com o objetivo de facilitar a utilização das diferentes ferramentas, a leitura de um TEDS é efetuada seguindo os mesmos princípios. Nesse sentido, o funcionamento desta ferramenta para realizar a leitura de um TEDS segue a mesma sequência de utilização da ferramenta *User's Transducer Name* TEDS, descrita na subsecção 5.3.1.1.

#### 5.3.4.2 *User's Transducer Name* TEDS

O ficheiro 'XdrName\_TEDS.teds' define um *User's Transducer Name* TEDS, cujo objetivo é o de armazenar o nome pelo qual o sistema reconhece o módulo de saída. Como no módulo de entrada foi definido um TEDS do mesmo tipo, nesta subsecção apenas será feita uma breve descrição da estrutura de dados de ambos os ficheiros.

A identificação do TEDS tem como objetivo a sua descrição. A aplicação define, por regra, os valores corretos para a Identificação do TEDS. O formato do TEDS define o conteúdo do bloco de dados. Na aplicação foi definida a opção *User-defined*. O nome do módulo IEEE1451.0 é "Tim". Na aplicação o utilizador define este campo através de uma caixa de texto. A criação do *User's Transducer Name* TEDS termina com o cálculo do tamanho do TEDS e do *checksum*.

O tamanho total deste TEDS é de 16 octetos e o valor do *checksum* é de 65187. Estes dois resultados são calculados e convertidos pela aplicação no momento de criação do TEDS. Ambos os ficheiros TEDSs do módulo IEEE 1451.0 foram definidos. Desta forma, foi concluído o teste relativo à definição dos ficheiros utilizados neste módulo de laboratório. Importa, de seguida, descrever a criação dos ficheiros que permitem mapear os TEDSs com o módulo IEEE 1451 definido na FPGA.

### 5.3.5 Ficheiro *Map Table*

O ficheiro *Map Table* define o mapeamento dos TEDSs com o módulo IEEE 1451. Assim, os TEDSs do TIM e dos módulos de laboratório, interligados ao módulo IEEE 1451 pelos *TransducerChannels*, são associados a uma determinada posição de memória aquando da implantação numa FPGA. No cenário adotado, é necessário a criação de um ficheiro *Map Table* com a descrição para ambas as configurações.

Na configuração 1 são utilizados os três módulos de laboratório. Desta forma, a criação do *Map Table* deve conter os TEDSs do TIM, bem como do módulo de entrada, de saída e do SMCM. Em relação ao TIM foram definidos dois TEDS: *Meta-TEDS* (código de acesso é um) e *User's Transducer Name TEDS* (código de acesso é doze). O *Meta-TEDS* é definido na posição de memória zero e o *User's Transducer Name TEDS* na posição de memória um. Tendo em consideração o número de identificação (0), a conversão destes valores para o formato TLV origina o resultado representado na tabela 5.9.

**Tabela 5.9: Definição do *Map Table* - TIM.**

	<b>Campo</b>	<b>Tipo de dados</b>	<b>Octetos</b>	<b>Valor</b>	<b>Valor (hexa)</b>
<b>TIM</b>	Número de identificação	UInt16	2	0	00.00
	Tamanho	UInt8	1	4	04
	Código de acesso ao <i>Meta-TEDS</i>	UInt8	1	1	01
	Posição de memória	UInt8	1	0	00
	Código de acesso ao <i>User's Transducer Name TEDS</i>	UInt8	1	12	0C
	Posição de memória	UInt8	1	1	01

Além do TIM, é ainda necessário interligar os TEDSs definidos por cada módulo de laboratório. O TC1 interliga com o *TransducerChannel TEDS* no módulo de entrada. O TC2 interliga com o *TransducerChannel TEDS* no módulo de saída. O TC3 interliga com o *TransducerChannel TEDS* e o *Manufacturer-defined TEDS* no módulo SMCM. A criação do *Map Table* termina com o cálculo do tamanho do TEDS e do *checksum*. Estes dois campos são calculados e convertidos, cujo resultado final é representado na tabela 5.10.

**Tabela 5.10: Formato final do *Map Table* - Configuração 1.**

	<b>Campo</b>	<b>Tipo de dados</b>	<b>Octetos</b>	<b>Valor</b>	<b>Valor (hexa)</b>
<b>TIM</b>	Tamanho do <i>Map Table</i>	UInt32	4	26	00.00.00.1A
	Número de identificação	UInt16	2	0	00.00
	Tamanho	UInt8	1	4	04
	Código de acesso ao <i>Meta-TEDS</i>	UInt8	1	1	01
	Posição de memória	UInt8	1	0	00
	Código de acesso ao <i>User's Transducer Name TEDS</i>	UInt8	1	12	0C
	Posição de memória	UInt8	1	1	01
	<b>TC1</b>	Número de identificação	UInt16	2	1
Tamanho		UInt8	1	2	02
Código de acesso ao		UInt8	1	3	03

	<i>TransducerChannel</i> TEDS				
	Posição de memória	UInt8	1	2	02
	Número de identificação	UInt16	2	2	00.02
	Tamanho	UInt8	1	2	02
<b>TC2</b>	Código de acesso ao <i>TransducerChannel</i> TEDS	UInt8	1	3	03
	Posição de memória	UInt8	1	3	03
	Número de identificação	UInt16	2	3	00.03
	Tamanho	UInt8	1	4	04
	Código de acesso ao <i>TransducerChannel</i> TEDS	UInt8	1	3	03
<b>TC3</b>	Posição de memória	UInt8	1	4	04
	Código de acesso ao <i>Manufacturer-defined</i> TEDS	UInt8	1	128	80
	Posição de memória	UInt8	1	5	05
	<i>Checksum</i>	UInt16	2	65326	FF.2E

Na configuração 2 são utilizados três módulos de laboratório. Nesta configuração, o módulo SMCM não é reconfigurado pela infraestrutura, sendo substituído por uma reprodução do módulo de saída de 6-*Bit*. A definição do TIM na configuração 2 tem o mesmo valor da configuração 1.

O TC1 interliga com o *TransducerChannel* TEDS e o *User's Transducer Name* TEDS no módulo de entrada. O TC2 e o TC3 interligam com o *TransducerChannel* TEDS e o *User's Transducer Name* TEDS em cada um dos módulos de saída. A criação do *Map Table* termina com o cálculo do tamanho do TEDS e do *checksum*. Estes dois campos são calculados e convertidos, cujo resultado final é representado tabela 5.11.

**Tabela 5.11: Formato final do *Map Table* - Configuração 2.**

	<b>Campo</b>	<b>Tipo de dados</b>	<b>Octetos</b>	<b>Valor</b>	<b>Valor (hexa)</b>
	Tamanho do <i>Map Table</i>	UInt32	4	30	00.00.00.1E
	Número de identificação	UInt16	2	0	00.00
	Tamanho	UInt8	1	4	04
	Código de acesso ao <i>Meta</i> -TEDS	UInt8	1	1	01
<b>TIM</b>	Posição de memória	UInt8	1	0	00
	Código de acesso ao <i>User's Transducer Name</i> TEDS	UInt8	1	12	0C
	Posição de memória	UInt8	1	1	01
	Número de identificação	UInt16	2	1	00.01
	Tamanho	UInt8	1	4	04
	Código de acesso ao <i>Meta</i> -TEDS	UInt8	1	1	01
<b>TC1</b>	Posição de memória	UInt8	1	2	02
	Código de acesso ao <i>User's Transducer Name</i> TEDS	UInt8	1	12	0C
	Posição de memória	UInt8	1	3	03
	Número de identificação	UInt16	2	2	00.02
	Tamanho	UInt8	1	4	04
	Código de acesso ao <i>Meta</i> -TEDS	UInt8	1	1	01
<b>TC2</b>	Posição de memória	UInt8	1	4	04
	Código de acesso ao <i>User's Transducer Name</i> TEDS	UInt8	1	12	0C

	Posição de memória	UInt8	1	5	05
	Número de identificação	UInt16	2	3	00.03
	Tamanho	UInt8	1	4	04
	Código de acesso ao <i>Meta</i> -TEDS	UInt8	1	1	01
<b>TC3</b>	Posição de memória	UInt8	1	6	06
	Código de acesso ao <i>User's</i> <i>Transducer Name</i> TEDS	UInt8	1	12	0C
	Posição de memória	UInt8	1	7	07
	<i>Checksum</i>	UInt16	2	65397	FF.75

Para a definição de um ficheiro *Map Table* será utilizada a ferramenta *Config Map Table*. A interface da ferramenta está dividida em duas partes: TIM e o grupo de *TransducerChannels*. Em relação ao TIM foram definidos dois TEDS: *Meta*-TEDS (código de acesso é um) e *User's* *Transducer Name* TEDS (código de acesso é doze). Na aplicação é necessário adicionar os dois TEDS na estrutura associada ao TIM, através do botão mais, como é ilustrado na figura 5.19.

**Figura 5.19: Interface da ferramenta *Config Map Table* - TIM.**

Em cada um dos TEDS é necessário configurar o código de acesso. Esta definição é feita com base numa caixa de combinação. Desta forma, para cada opção selecionada pelo utilizador é apresentada uma mensagem com o respetivo nome do TEDS.

Na estrutura de dados do grupo dos *TransducerChannels* é necessário adicionar o número de *TransducerChannels* do cenário adotado, neste caso três. Esta ação é efetuada através do botão mais. De seguida, com base na configuração 1, é necessário adicionar e configurar os TEDSs de cada *TranducerChannel*, como é ilustrado figura 5.20. Além do tamanho do *Map Table* e do *checksum*, os campos número de identificação e posição de memória são calculados pela aplicação.

A ferramenta *Config Map Table* permite ainda a leitura de um ficheiro *Map Table*. Para a leitura do *Map Table* é necessário que o utilizador selecione o botão *Search file*, que disponibiliza uma janela de pesquisa para escolher o ficheiro pretendido. De seguida, o utilizador executa o botão *Submit* que importa o ficheiro selecionado na ferramenta *Config Map Table*, como é apresentado na figura 5.21.

**Figura 5.20: Interface da ferramenta *Config Map Table* – *TransducerChannels*.**

**Figura 5.21: Interface da ferramenta *Config Map Table* – leitura do ficheiro.**

A importação do ficheiro realiza duas ações principais: i) converte os dados do ficheiro binário para hexadecimal e, ii) valida se o ficheiro selecionado foi criado de acordo com a norma IEEE 1451.0, tendo em consideração o valor do *checksum*, bem como a estrutura TLV dos diferentes tuplos. Se o ficheiro for considerado inválido é apresentada uma mensagem de erro e os dados não são importados. Se o ficheiro for válido os dados são importados pela ferramenta com o objetivo de criar todos os *TransducerChannels* necessários e de configurar os diferentes TEDSs. No entanto, apesar da validade do ficheiro existe a possibilidade do valor de um dos campos não estar configurado corretamente (e.g. estar definido fora dos limites, estar definido com um diferente tipo de dados, entre outros). Nesse momento, a importação do ficheiro termina e é apresentada no topo da página uma mensagem de erro com o motivo dessa paragem.

A alteração de um valor do *Map Table* é um processo igualmente simples. Após a importação do ficheiro, o utilizador tem permissão para alterar qualquer valor definido pelo *Map Table*, com a exceção dos campos criados automaticamente pela ferramenta (e.g. *checksum*, Tamanho do TEDSs, posição de memória, entre outros). Desta forma, a ferramenta *Config Map Table* disponibiliza, de forma intuitiva, vários botões para adicionar, subtrair ou remover



os *TransducerChannels* e TEDSs. Conforme é apresentado pela figura 5.22, no TIM o utilizador pode adicionar (botão mais), subtrair (botão menos) o último TEDS ou remover (botão cruz) um TEDS diretamente. A interface define ainda o grupo dos *TransducerChannels*. De acordo com a configuração necessária, o utilizador pode adicionar ou subtrair cada um dos *TransducerChannels*. À semelhança do que é possível no TIM, em cada *TransducerChannel* o utilizador tem permissão para adicionar, subtrair o último TEDS ou remover um TEDS diretamente. No final das alterações é essencial voltar a executar o botão *Build*, para validar os novos dados, e o botão *Create*, para criar novamente o ficheiro.

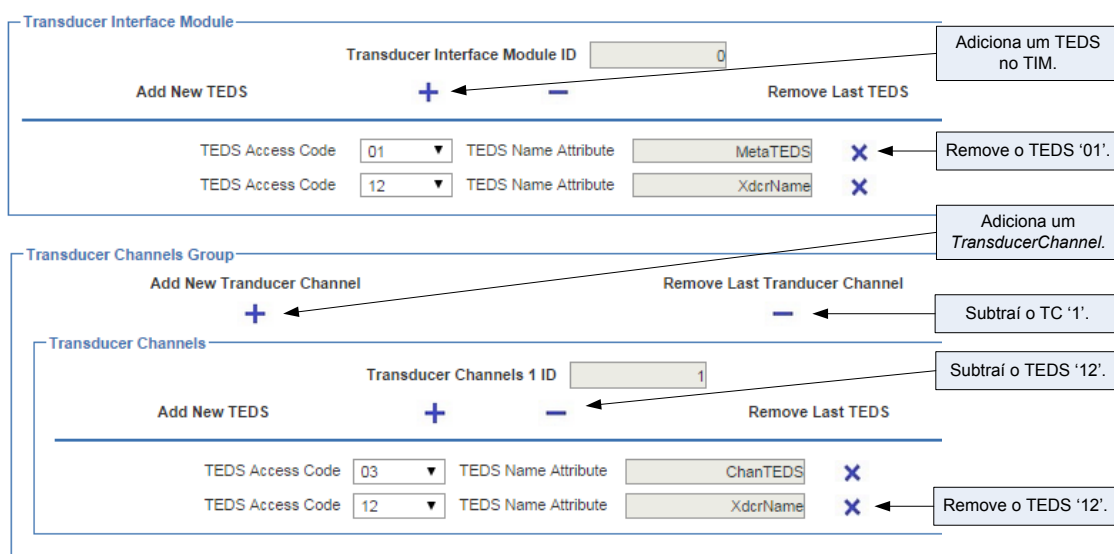


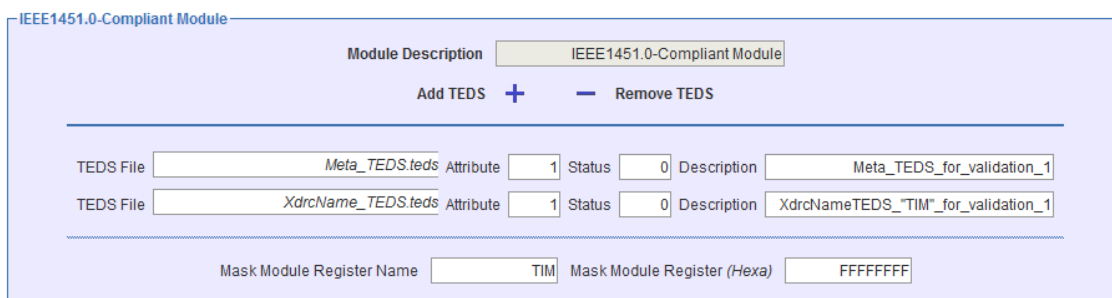
Figura 5.22: Interface da ferramenta *Config Map Table* – opções de alteração.

Ambos os ficheiros *Map Table* foram definidos. Desta forma, foi concluído o teste relativo à criação destes ficheiros pela aplicação. Importa, de seguida, descrever a criação dos ficheiros que permitem a interligação dos módulos de laboratório com a infraestrutura do laboratório remoto.

### 5.3.6 Ficheiro de *Configuração*

O ficheiro de *Configuração* define a interligação dos diferentes módulos de laboratório com infraestrutura do laboratório remoto. São definidas todas as regras necessárias para reconfigurar o projeto de um laboratório remoto. Para a definição destes ficheiros será utilizada a ferramenta *Configure Weblab Modules Connections*.

No cenário é necessário a criação deste ficheiro para ambas as configurações. O ficheiro de *Configuração* é composto pela definição dos dados atribuídos ao módulo IEEE 1451.0, bem com a cada um dos módulos de laboratório. A descrição do TIM (IEEE 1451.0-Compliant Module) é atribuída de forma automática, como é ilustrado na figura 5.23.



**Figura 5.23: Interface da ferramenta *Configure Weblab Modules Connections* – TIM.**

De seguida, é necessário definir os vários TEDSs definidos no TIM. No TIM foram definidos dois TEDS: *Meta-TEDS* e *User's Transducer Name TEDS*. Em cada um é necessário configurar o nome do ficheiro que o define, o campo atributo, o campo estado e adicionar uma descrição do TEDS. Para definir o nome do ficheiro do TEDS é necessário clicar no campo *TEDS File*, que disponibiliza uma janela de pesquisa para selecionar o ficheiro pretendido. O campo atributo indica se o TEDS é suportado pelo TC, se pode ser alterado pelo TIM ou pelo TC, se o conteúdo atual é válido e, se o TEDS está embutido no TIM ou localizado remotamente (virtual). O campo estado indica a condição atual do TEDS. Ambos os campos são definidos pela norma IEEE 1451.0. O campo atributo é definido com o valor um porque o TEDS não é suportado pelo TC. O campo estado é definido com o valor zero porque o tamanho da memória necessária para alocar o TEDS não é suficiente. Por fim, é definido o nome do módulo e a respetiva máscara em hexadecimal.

Após a configuração do TIM, é necessário configurar os diferentes módulos de laboratório. No cenário adotado são utilizados três módulos de laboratório. Inicialmente, para cada um deles, é necessário definir o nome do módulo principal. Esta descrição é da responsabilidade do utilizador (Input\_8bits\_main). Conforme é apresentado na figura 5.24, para além da descrição, a definição de cada módulo de laboratório é composta pelas estruturas: i) *TransducerChannel*, ii) *external wires* e, iii) *HDL verilog files*.

A estrutura *TransducerChannel* define os TCs interligados em cada módulo de laboratório. Em cada TC é necessário definir os seus diferentes TEDSs. No módulo de entrada é definido um *TransducerChannel* TEDS. Nesse contexto, é necessário configurar o nome do ficheiro que o define, o campo atributo, o campo estado e adicionar a respetiva descrição do TEDS.

De seguida, é definido o nome do módulo e a respetiva máscara em hexadecimal. Associado a cada TC está um ficheiro HDL que define a interligação do módulo de laboratório com o módulo IEEE 1451.0. De forma genérica, neste ficheiro são definidas todas as tarefas associadas a este TC. Para definir o nome do ficheiro HDL é necessário clicar no campo *Task Location*, que disponibiliza uma janela de pesquisa para selecionar o ficheiro pretendido. Para além da definição deste ficheiro, são configuradas na ferramenta as tarefas executadas por cada TC. Posteriormente é realizada a configuração dos sinais usados para interligar os TCs com os módulos de laboratório. O número de fios é controlado pelo utilizador. Em cada

ligação é essencial definir o nome, o tamanho e a direção do fio. Por fim, são ainda definidas as variáveis de cada TC.

The screenshot shows the 'Configure Weblab Modules Connections' interface. At the top, the 'Main WebLab HDL Module Name' is set to 'Input\_8bits\_main'. Below this, there are buttons to 'Add Tranducer Channel' and 'Remove Tranducer Channel'. The 'Transducer Channel 1' section contains an 'Add TEDS' and 'Remove TEDS' button. The TEDS configuration includes fields for 'TEDS File' (Input\_8bits\_TC\_Channel.teds), 'Attribute' (1), 'Status' (0), and 'Description' (TC\_TEDS\_input8bits). Below this, there are fields for 'Mask Register Name' (input8bits), 'Mask Register (Hexa)' (FFFFFFF), and 'Task Location (.vh)' (Input\_8bits\_tasks.vh). A section for 'Select the Associated Tasks' includes checkboxes for 'Start', 'Stop', 'Read' (checked, tc1), 'Write', 'Init' (checked, tc1), 'Update' (checked, tc1), and 'Event'. Below this, there are 'Add Wire' and 'Remove Wire' buttons. The wire configuration section shows three wires: 'en' (width 1, direction output), 'rst' (width 1, direction output), 'in' (width 7, direction input), 'run' (width 1, direction output), and 'end' (width 1, direction input). Below this, there are 'Add Variable' and 'Remove Variable' buttons. The variable configuration section shows 'Variable Name' (state\_tc1) and 'Value' (0). The 'External Wires' section has a note: '(Note: The UCF file must be in accordance with these wires.)' and buttons to 'Add External Wire' and 'Remove External Wire'. It shows two external wires: 'data\_in\_tc1' (width 7, direction input) and 'clk' (width 1, direction input, with 'Clk' checked and 'Event' unchecked). The 'HDL Verilog Files' section has buttons to 'Add File' and 'Remove File', and a field for 'Weblab Module File Location (.v)' (Input\_8bits\_main.v).

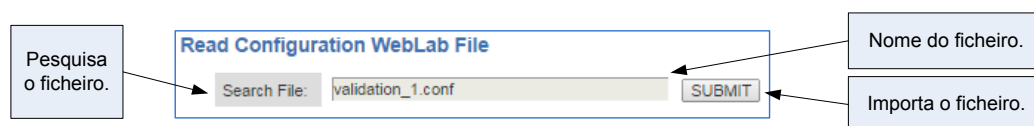
**Figura 5.24: Interface da ferramenta *Configure Weblab Modules Connections* – módulo de laboratório.**

A estrutura *external wires* define os fios externos de entrada e saída de cada módulo de laboratório. Estes fios externos são sinais usados para interligar os diferentes módulos de laboratório com a FPGA. O número de fios externos é controlado pelo utilizador. Em cada ligação é necessário definir o nome, o tamanho e a direção do fio.

Por fim, a estrutura *HDL verilog files* define os ficheiros HDL que descrevem o módulo de laboratório. O número de ficheiros é controlado pelo utilizador. Em cada entrada é necessário definir o nome do ficheiro HDL. Para a sua definição é necessário clicar no campo *Weblab Module File Location*, que disponibiliza uma janela de pesquisa para selecionar o ficheiro pretendido.

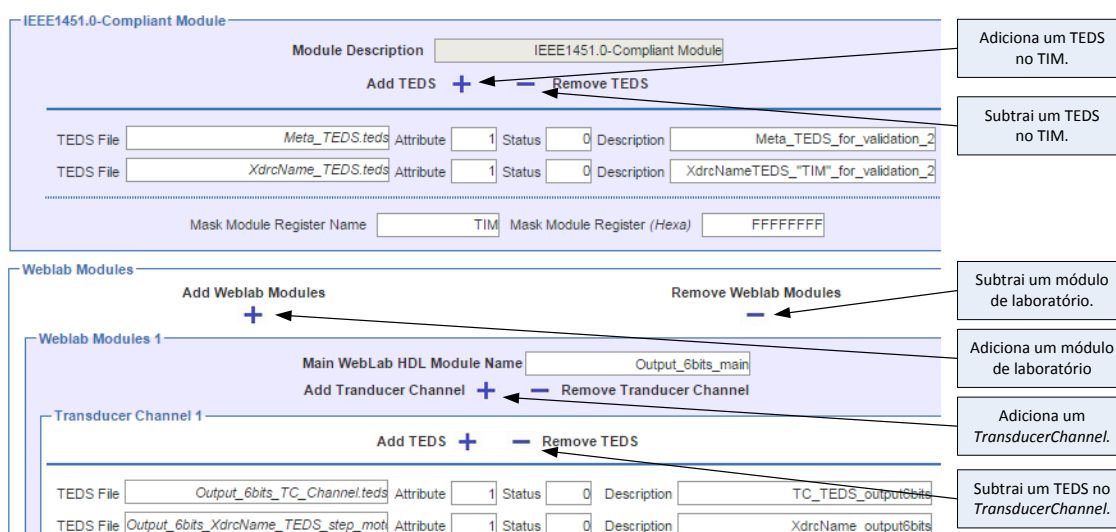
A ferramenta *Configure Weblab Modules Connections* permite ainda a leitura de um ficheiro de *Configuração*. Para isso, é necessário que o utilizador selecione o botão *Search file*, que disponibiliza uma janela de pesquisa para escolher o ficheiro pretendido. De seguida, o utilizador executa o botão *Submit* que importa o ficheiro selecionado na ferramenta *Configure Weblab Modules Connections*, como é apresentado na figura 5.25.

A importação do ficheiro verifica se este foi criado de acordo com as regras. Se o ficheiro for considerado inválido é apresentada uma mensagem de erro e os dados não são importados. Se o ficheiro for válido os dados são importados pela ferramenta com o objetivo de definir as várias configurações do TIM e dos diferentes módulos de laboratório. No entanto, apesar da validade do ficheiro existe a possibilidade do valor de um dos campos não estar configurado corretamente (e.g. estar definido fora dos limites, estar definido com um diferente tipo de dados, entre outros). Nesse momento, a importação do ficheiro termina e é apresentada no topo da página uma mensagem de erro com o motivo dessa paragem.



**Figura 5.25:** Interface da ferramenta *Configure Weblab Modules Connections* – leitura do ficheiro.

A alteração de um valor no ficheiro de *Configuração* é um processo igualmente simples. Após a importação do ficheiro, o utilizador tem permissão para alterar qualquer valor definido, com a exceção dos campos criados automaticamente pela ferramenta (e.g. descrição do TIM). Desta forma, a ferramenta *Configure Weblab Modules Connections* disponibiliza, de forma intuitiva, vários botões para adicionar (botão mais) ou subtrair (botão menos) os módulos de laboratório, os *TransducerChannels*, os TEDSs, entre outros. Conforme é apresentado pela figura 5.26, no TIM o utilizador pode adicionar ou subtrair o último TEDS.



**Figura 5.26:** Interface da ferramenta *Configure Weblab Modules Connections* – opções de alteração.

De seguida, a interface define um conjunto de módulos de laboratório. De acordo com a configuração necessária, o utilizador pode adicionar ou subtrair cada um dos módulos de laboratório. Em cada um dos módulos é possível adicionar ou subtrair vários *TransducerChannels*.

À semelhança do que é possível no TIM, em cada *TransducerChannel* o utilizador tem permissões para adicionar ou subtrair o último TEDS, os fios de ligação e as diferentes variáveis. Na estrutura do módulo de laboratório é ainda possível adicionar ou subtrair os fios externos e os ficheiros HDL. No final das alterações é essencial voltar a executar o botão *Build*, para validar os novos dados, e o botão *Create*, para criar novamente o ficheiro.

Após a descrição do cenário adotado e da realização das tarefas de leitura e escrita dos vários ficheiros utilizados, é importante realizar uma avaliação qualitativa da aplicação, tendo em consideração os objetivos inicialmente propostos.

## 5.4 Avaliação qualitativa

A avaliação qualitativa da aplicação *Web* pretende apresentar as vantagens e desvantagens face à definição manual dos diferentes ficheiros utilizados na criação do cenário adotado, tendo em consideração a leitura e escrita dos mesmos.

A principal vantagem na leitura e escrita manual dos ficheiros reside na facilidade de acesso independentemente do lugar, existindo apenas a necessidade de um computador para realizar os devidos cálculos e conversões dos dados utilizados. Por outro lado, a utilização da aplicação obriga à existência de uma ligação à *Internet*.

Uma das dificuldades sentidas em ambos os casos é a falta de apoio relacionado com a norma IEEE 1451.0. Apesar da aplicação identificar os diferentes campos obrigatórios e opcionais de cada TEDS, a complexidade para a sua configuração exige ao utilizador conhecimentos da norma IEEE 1451.0, relacionados com as funções de cada tuplo.

A criação dos ficheiros com recurso à aplicação garante outras facilidades, mais concretamente, devido à automatização de todas as conversões e cálculos na preparação dos respetivos ficheiros. No processo manual, a realização das conversões e cálculos durante a definição dos TEDSs são duas das principais desvantagens, devido à sua estrutura subjacente. Para além disso, a possibilidade de erro durante a realização destes cálculos e conversões de forma manual é superior à do processo com a aplicação, cuja probabilidade de erro é nula. Desta forma, os ficheiros criados pela aplicação respeitam a estrutura TLV definida na norma IEEE 1451.0, bem como garantem a integridade dos dados através do *checksum* definido em cada TEDS.

Outra possível comparação reside na leitura de um ficheiro já criado. A leitura deste ficheiro de forma manual exige a utilização de ferramentas externas para ler os respetivos dados binários. Aliada a esta desvantagem, existe ainda a dificuldade na alteração de um

determinado valor. No processo manual é necessário definir o novo valor e realizar novamente os cálculos e conversões para criar o ficheiro. No processo com recurso à aplicação apenas é necessário importar novamente o ficheiro e configurar o novo valor no respetivo campo. De seguida, os cálculos e conversões são executados novamente pela aplicação, garantido assim facilidade na alteração de um determinado ficheiro.

Com base nas considerações apresentadas nos parágrafos anteriores, a tabela 5.12 resume as vantagens e desvantagens na leitura e escrita de um ficheiro com a aplicação face ao processo manual.

**Tabela 5.12: Vantagens e desvantagens na utilização da aplicação desenvolvida face ao processo manual.**

	<b>Processo manual</b>	<b>Aplicação desenvolvida</b>
<b>Vantagens</b>	Facilidade de acesso independente do lugar.	Automatização das conversões.
		Automatização dos cálculos.
		Respeito pela estrutura TLV.
		Garantia da integridade dos dados.
		Rapidez na leitura de um TEDS.
<b>Desvantagens</b>		Facilidade na alteração de um campo.
	Acesso a documentação detalhada	relacionada com a norma IEEE 1451.0.
	Dificuldade na realização das conversões.	Necessidade de ligação à <i>Internet</i> .
	Dificuldade na realização dos cálculos.	Necessidade de perceber o funcionamento da aplicação.
	Maior possibilidade de erro.	
	Dificuldade na alteração de um campo.	
	Utilização de ferramentas externas.	

## 5.5 Sumário

Neste capítulo foi apresentada a verificação experimental da aplicação desenvolvida no âmbito desta dissertação. Após a descrição dos principais objetivos, a verificação foi realizada com base num cenário que utiliza duas configurações diferentes e três módulos de laboratório. De forma a tirar partido da aplicação, foram criados os TEDSs utilizados pelos módulos de laboratório e os ficheiros *Map Table* e de *Configuração* do laboratório segundo as configurações apresentadas no cenário adotado. Por fim, foi realizada uma avaliação qualificativa da aplicação, através de uma análise funcional da leitura e escrita dos diferentes ficheiros.

## 6 Conclusão

Neste capítulo é resumido o trabalho realizado durante o desenvolvimento deste projeto. São apresentados os objetivos alcançados, bem como as principais linhas de orientação seguidas para a elaboração desta dissertação. De seguida, são identificadas as principais dificuldades e limitações do trabalho, bem como apresentadas futuras perspectivas de desenvolvimento.

### 6.1 Resumo

Nesta dissertação foi analisado o contexto atual do ensino à distância, tendo em conta as suas duas componentes fundamentais: a teórica e a prática. O estudo sobre cada uma das componentes está obrigatoriamente relacionado, pois na transmissão da parte teórica é importante que esta se relacione com o mundo real, isto é, com a componente prática.

A componente prática é desenvolvida através de exercícios práticos ou do trabalho experimental. No entanto, apesar da importância dos exercícios práticos para permitir uma maior autonomia aos alunos, é no trabalho experimental que estes são provocados e motivados para resolverem os problemas reais. Foi realizada uma análise e estudo dos diferentes ambientes laboratoriais, com particular enfoque nos laboratórios remotos. Além disso, foi também apresentada a sua arquitetura e referenciados alguns exemplos de laboratórios remotos. Foram descritos os seus problemas atuais e apresentado um protótipo desenvolvido no âmbito de um trabalho de doutoramento [1] para resolver estas limitações.

Este protótipo propõe o desenvolvimento de laboratórios remotos reconfiguráveis, cuja arquitetura é baseada na norma IEEE 1451.0 e na tecnologia de FPGAs. O estudo deste protótipo teve em consideração os principais aspetos de desenvolvimento, destacando-se: i) a infraestrutura do laboratório remoto, onde estão localizados o TIM e o NCAP, e ii) o servidor do laboratório remoto, que é acedido através da *RecTool*.

Para uma melhor compreensão do funcionamento deste protótipo foi também realizado um estudo sobre a norma IEEE 1451.0. Foi apresentada a sua arquitetura, composta por um

processador de rede (NCAP) interligado a um módulo para a ligação de transdutores inteligentes (TIM), e as suas principais funcionalidades. Devido à relevância para esta dissertação, foi descrita com particular enfoque a estrutura dos vários TEDSs desta norma. No entanto, devido à dificuldade de criação destes ficheiros por parte do utilizador foi necessário o desenvolvimento de uma aplicação *Web* para a leitura e escrita dos TEDSs, assim como dos ficheiros que interligam os módulos de laboratório com a infraestrutura do laboratório remoto.

Para o desenvolvimento desta aplicação, e tendo em consideração os requisitos necessários, foi desenhada uma arquitetura com base no modelo cliente-servidor. Foram apresentadas as diferentes interfaces de cada ferramenta da aplicação, bem como as suas principais funcionalidades. De seguida, foram mencionados os aspetos de desenvolvimento da aplicação, fazendo-se referência aos pormenores técnicos utilizados.

Por fim, foi descrito um cenário prático, escolhido para realizar a verificação experimental da aplicação desenvolvida. Com base neste cenário, foram criados, através da aplicação, os vários TEDSs utilizados nos módulos de laboratório, bem como os ficheiros *Map Table* e de *Configuração* que interligam estes módulos com a infraestrutura do laboratório remoto. No final desta criação foi realizada uma avaliação qualitativa da aplicação, referindo-se as suas vantagens e desvantagens em relação à definição manual dos ficheiros utilizados. No entanto, durante o desenvolvimento desta aplicação, foram igualmente encontradas várias dificuldades e limitações que serão descritas na subsecção seguinte.

## 6.2 Principais dificuldades e limitações

Durante a realização desta dissertação foram encontradas várias dificuldades subjacentes à interpretação da norma IEEE 1451.0. Mais concretamente, durante o estudo desta norma foram detetadas algumas contradições na definição dos vários TEDSs, pelo que foi necessário analisar os exemplos práticos para prosseguir o desenvolvimento da aplicação. Infelizmente, por se tratar de uma norma relativamente recente, existem poucos exemplos práticos, com a exceção dos apresentados na própria norma.

Outra dificuldade inerente reside na especificação dos TEDSs. Cada TEDS tem uma estrutura única e variável, exigindo um grande rigor para os definir. Para além da sua estrutura, a definição de cada TEDS tem um grau de dificuldade elevado, pois é necessário a realização de vários cálculos e conversões. Para isso, foi necessário recorrer de algumas ferramentas externas para a conversão de certos tipos de dados, nomeadamente para a representação em vírgula flutuante de acordo com a norma IEEE 754.

Por fim, importa ainda referir a dificuldade para a compreensão do protótipo previamente desenvolvido, devido ao uso de linguagens, tecnologias e sistemas não usuais no curso de Engenharia Informática. O conhecimento necessário foi obtido através do estudo de artigos relacionados com este protótipo, bem como a visualização de vários vídeos sobre a reconfiguração de laboratórios remotos.



Apesar das dificuldades e limitações apresentadas, a aplicação desenvolvida é um bom contributo para o apoio de laboratórios remotos reconfiguráveis baseados na norma IEEE 1451.0, existindo ainda a possibilidade de ser melhorada no futuro.

### **6.3 Perspetivas de desenvolvimento**

A aplicação desenvolvida permite a leitura e escrita de TEDSs, com base na norma IEEE 1451.0. Apesar do cumprimento dos objetivos inicialmente propostos desta dissertação, existem algumas melhorias que podem ser implementadas no futuro, nomeadamente:

- Desenvolver novas ferramentas de leitura escrita dos TEDSs opcionais mencionados na norma IEEE 1451.0, com o objetivo de tornar a aplicação mais completa;
- Desenvolver um painel de suporte para ajudar os utilizadores durante a criação de um TEDS. Desta forma, em caso de dúvida sobre um determinado campo o utilizador teria a possibilidade de consultar a informação descrita na norma IEEE 1451.0;
- Desenvolver uma parte gráfica para facilitar a interligação dos módulos de laboratório com a infraestrutura do laboratório.



# Referências

- [1] R. Costa, Tese de Doutoramento, «*An IEEE1451.0 - compliant FPGA - based reconfigurable weblab*», Universidade de Coimbra, Coimbra, 2014.
- [2] R. Costa, G. Alves e M. Zenha-Rela, «*Embedding Instruments & Modules into an IEEE1451-FPGA-based Weblab Infrastructure*», International Journal of Online Engineering (iJOE), vol. 8, n. 3, agosto de 2012.
- [3] R. Costa, G. Alves e M. Zenha-Rela, «*Reconfigurable weblabs based on the IEEE1451 Std.*», Education Engineering (EDUCON), IEEE, 2010, pp. 1359–1366.
- [4] R. Costa, G. Alves e M. Zenha-Rela, «*A demo prototype of a reconfigurable IEEE1451. 0-compliant and FPGA-based weblab*», 2014.
- [5] R. Costa, G. Alves e M. Zenha-Rela, «*Reconfigurable IEEE1451-FPGA based weblab infrastructure*», International Conference on Remote Engineering and Virtual Instrumentation (REV'12), Bilbao, Spain, 2012.
- [6] R. Costa, G. Alves e M. Zenha-Rela, «*Using FPGAs to create a reconfigurable IEEE1451. 0-compliant weblab infrastructure*», IX Jornadas sobre Sistemas Reconfiguráveis, Instituto de Sistemas e Robótica, Universidade de Coimbra, 2013, pp. 11 – 14.
- [7] D. Wobschall, «*Networked sensor monitoring using the universal IEEE 1451 standard*», Instrumentation & Measurement Magazine, IEEE, vol. 11, n. 2, 2008, pp. 18–22.
- [8] R. Costa, Tese de Mestrado, «*Infra-estrutura laboratorial para experimentação remota*», Universidade do Porto - Faculdade de Engenharia da Universidade do Porto, Porto, 2003.
- [9] G. Teixeira, «*Introdução aos Conceitos de Educação, Ensino, Aprendizagem a Didáctica*», Ser Professor Universitário. (<http://www.serprofessoruniversitario.pro.br>).
- [10] Daniel e Ernestina Silva, «*Ensino clínico na formação em enfermagem*», Escola Superior de Enfermagem de Viseu - 30 anos, 2004, pp. 103 – 119.
- [11] S. Lopes, Tese de Mestrado, «*Laboratório de acesso remoto em Física*», Universidade de Coimbra, Coimbra, 2007.
- [12] D. Cardoso e E. Takahashi, «*Experimentação remota em actividades de ensino formal: um estudo a partir de periódicos Qualis A*», Revista Brasileira de Pesquisa em Educação em Ciências, vol. 11, n. 3, 2011, pp. 185 – 208.
- [13] L. Gomes e J. García-Zubía, «*Advances on remote laboratories and e-learning experiences*», Universidade de Deusto, Bilbao: Deusto - Publicações, 2007.
- [14] E. Monteiro e F. Boavida, «*Engenharia de Redes Informáticas*», 7ª ed. FCA - Editora de Informática, 2000.
- [15] G. Carnevali e G. Buttazzo, «*A virtual laboratory environment for real-time experiments*», Intelligent Components and Instruments for Control Applications (SICICA), Aveiro, Portugal, 2003, pp. 39 – 44.
- [16] V. J. Harward et al., «*The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories*», Proceedings of the IEEE, vol. 96, n. 6, junho de 2008, pp. 931 – 950.

- [17] I. Gustavsson et al., «*The VISIR Open Lab Platform 5.0 - an architecture for a federation of remote laboratories*», International Conference on Remote Engineering and Virtual Instrumentation (REV'11), Universidade de Brasov, Roménia, 2011, pp. 284 – 288.
- [18] R. Costa, G. Alves e M. Zenha-Rela, «*Reconfigurable weblabs based on the IEEE1451 Std.*», 1st IEEE Engineering Education 2010, Madrid, Espanha, 2010, pp. 1359 – 1366.
- [19] T. Filho et al., «*Descrição dos TEDS para o controle de motores de passo em conformidade com o padrão IEEE 1451*», XVIII Congresso Brasileiro de Automática, setembro de 2010, pp. 4761 – 4766.
- [20] F. Tani, «*Proposta de desenvolvimento de transdutores inteligentes baseados na norma IEEE 1451 aplicados a redes Lonworks*», Escola Politécnica da Universidade de São Paulo, São Paulo, 2006.
- [21] E. Y. Song e K. Lee, «*Understanding IEEE 1451-Networked smart transducer interface standard-What is a smart transducer?*», Instrumentation & Measurement Magazine, IEEE, vol. 11, n. 2, 2008, pp. 11–17.
- [22] IEEE Instrumentation and Measurement Society, TC-9 Committee on Sensor Technology, Institute of Electrical and Electronics Engineers, American National Standards Institute, e IEEE-SA Standards Board, «*IEEE standard for a smart transducer interface for sensors and actuators common functions, communication protocols, and transducer electronic data sheet (TEDS) formats*». New York, NY: Institute of Electrical and Electronics Engineers, 2007.
- [23] R. Jesus, «*MATIW 1451 - Monitoramento e Accionamento de Transdutores Inteligentes através da Web (Padrão IEEE 1451)*», Universidade Estadual Paulista Júlio de Mesquita Filho, Ilha Solteira, 2007.
- [24] N. US Department of Commerce, «*IEEE 1451 Smart Transducer Interface Standards*». (<http://www.nist.gov/el/isd/ieee/ieee1451.cfm>).
- [25] L. Torri, «*A norma IEEE 1451 aplicada a redes heterogêneas de sensores sem \_fio*», Universidade Federal de Santa Catarina, Florianópolis, 2008.
- [26] D. Wobschall, «*IEEE 1451—a universal transducer protocol standard*», Autotestcon, IEEE, 2007, pp. 359–363.
- [27] S. Manda e D. Gurkan, «*IEEE 1451.0 compatible TEDS creation using. NET framework*», Sensors Applications Symposium, IEEE, New Orleans, LA, USA, 2009, pp. 281–286.
- [28] W. Liu, Tese de Mestrado, «*Design of TEDS writer, reader and testing system for transducer interface modules based on the IEEE 1451 standard*», University of New York at Buffalo, 2006.
- [29] J. Valade, «*PHP & MySQL for dummies*», 4th ed. Hoboken, NJ: Wiley Pub., Inc, 2010.
- [30] N. Kew, «*The Apache modules book: application development with Apache*». Upper Saddle River, N.J.; London: Prentice Hall, 2007.
- [31] «*The Google Geocoding API*», Google Maps API Web Services. (<https://developers.google.com/maps/documentation/geocoding>).
- [32] R. Costa et al., «*Peers' evaluation of a reconfigurable IEEE1451. 0-compliant and FPGA-based weblab*», Experiment@ International Conference (exp. at'13), 2013, pp. 83–88.

# Anexo A

## Estrutura dos TEDSs

Este anexo apresenta a estrutura dos vários TEDSs que a aplicação desenvolvida permite criar e interpretar para serem utilizados pela *RecTool* na configuração de laboratórios remotos.

Na tabela a.1 são apresentados os campos, obrigatórios ou opcionais, que definem o *Meta-TEDS* de acordo com a norma IEEE 1451.0 [22]. Além disso, é ainda descrito o identificador de cada campo, a sua descrição, e o seu respetivo tamanho.

Tabela A.1: Estrutura do *Meta-TEDS*.

ID do campo	Nome do Campo	Descrição	Tipo de Dados	N Octetos
—		Tamanho	UInt32	4
0-2	—	Reservado <sup>40</sup>	—	—
3	TEDSID	TEDS <i>Identification Header</i>	UInt8	4
4	UUID	Identificador global exclusivo	UUID	10
5-9	—	Reservado	—	—
Informação relacionada com o tempo				
10	OholdOff	Tempo limite de operação	Float32	4
11	SHoldOff	Tempo limite de para acessos lentos	Float33	4
12	TestTime	Tempo para realizar testes automáticos	Float34	4
13	MaxChan	Número de TCS implementados	UInt16	2
14	CGroup	Bloco de informação sobre o grupo de controlo	—	—
20	GrpType	Tipo de grupo de controlo	UInt8	1
21	MemList	Lista de membros do grupo de controlo	<i>array of</i> UInt16	NTc
15	VGroup	Bloco de informação sobre o grupo de vetor	—	—
20	GrpType	Tipo grupo de vetor	UInt8	1
21	MemList	Lista de membros do grupo de vetor	<i>array of</i> UInt16	NTv
16	GeoLoc	Bloco de informação sobre a localização geográfica	—	—
24	LocEnum	Informação da localização	UInt8	1
20	GrpType	Tipo grupo de vetor	UInt8	1
21	MemList	Lista de membros do grupo de vetor	<i>array of</i> UInt16	NTv

<sup>40</sup> Os campos descritos como “reservados” são guardados para futuras implementações da norma IEEE 1451. Estes campos não são usados pelos fabricantes.

17	<i>Proxies</i>	Bloco de informação sobre o grupo de TCs <i>proxy</i>	—	—
22	ChanNum	Número de TCs <i>proxy</i>	UInt16	1
23	Organiz	Organização dos TCs <i>proxy</i>	UInt8	1
21	MemList	Lista de membros do grupo de TCs <i>proxy</i>	<i>array of</i> UInt16	NTP
18 - 19	—	Reservado	—	—
25-127	—	Reservado	—	—
128- 255	—	Aberto aos fabricantes <sup>41</sup>	—	—
—		<i>Checksum</i>	UInt16	2

Os campos obrigatórios do *META-TEDS* são:

- **Tamanho:** é utilizado para definir o número de octetos que o TEDS descreve;
- **TEDSID:** contém a estrutura de dados definida no início de todos os TEDSs da norma IEEE 1451.0;
- **UUID:** guarda a identificação única do TIM. É formado por 10 octetos, formados por 4 subcampos associados ao TEDS: localização, fabricante, ano e a data;
- **OholdOff:** define o menor período de tempo, em segundos, que o TIM pode ficar sem responder a um comando enviado pelo NCAP, até ser considerado uma falha na operação. Este valor é formado por 4 octetos e é representado em vírgula flutuante de acordo com a norma IEEE 754;
- **TestTime:** define o maior tempo, em segundos, que o TIM tem para executar um teste automático. O seu valor é constituído por 4 octetos e em vírgula flutuante;
- **MaxChan:** define o número de TCs que são implementados no TIM. Todos os TCs são numerados a partir de um. É formado por 2 octetos e deve ser um número inteiro positivo;
- **Checksum:** é o valor que garante a integridade do *META-TEDS*.

Na tabela a.2 são apresentados os campos, obrigatórios ou opcionais, que definem o *TransducerChannel* TEDS de acordo com a norma IEEE 1451.0 [22]. Além disso, é ainda descrito o identificador de cada campo, a sua descrição, e o seu respetivo tamanho.

**Tabela A.2: Estrutura do *TransducerChannel* TEDS.**

ID do campo	Nome do Campo	Descrição	Tipo de Dados	N Octetos
—		Tamanho	UInt32	4
0-2	—	Reservado	—	—
3	TEDSID	TEDS <i>Identification Header</i>	UInt8	4

<sup>41</sup> Os campos descritos como “aberto aos fabricantes” podem ser utilizados para implementar parâmetros que não são definidos na norma IEEE 1451.0.

4-9	—	Reservado	—	—
Informação relacionada com os TCs				
10	CalKey	Chave de calibração	UInt8	1
11	ChanType	Tipo de TCs	UInt8	1
12	PhyUnits	Unidades físicas	UNITS	11
50	UnitType	Enumeração das unidades físicas	UInt8	1
51	Radians	Valor para radianos	UInt8	1
52	SterRad	Valor para esterradiano	UInt8	1
53	Meters	Valor para metros	UInt8	1
54	Kilogram	Valor para quilogramas	UInt8	1
55	Seconds	Valor para segundos	UInt8	1
56	Amperes	Valor para amperes	UInt8	1
57	Kelvins	Valor para kelvins	UInt8	1
58	Moles	Valor para moles	UInt8	1
59	Candelas	Valor para candelas	UInt8	1
60	UnitsExt	Código de acesso para a extensão de unidades	UInt8	1
13	LowLimit	Valor operacional abaixo do limite	Float32	4
14	HiLimit	Valor operacional superior ao limite	Float32	4
15	OError	Grau de incerteza do pior caso	Float32	4
16	SelfTest	Testes automáticos	UInt8	1
17	MRange	Capacidade de intervalos múltiplos	UInt8	1
Informação relacionada com a conversão dos dados				
18	Sample		—	—
40	DatModel	Modelos de dados	UInt8	1
41	ModLenth	Tamanho do modelo de dados	UInt8	1
42	SigBits	<i>Bits</i> significativos do modelo de dados	UInt16	2
19	DataSet			
43	Repeats	Repetições máximas dos dados	UInt16	2
44	SOrigin	Series originais	Float32	4
45	StepSize	Series incrementais	Float32	4
46	SUnits	Unidades de series	UNITS	11
47	PreTrigg	Máximo de amostras de <i>triggers</i>	UInt16	2
Informação relacionada com o tempo				
20	UpdateT	Tempo de atualização dos TCs ( $t_u$ )	Float32	4
21	WSetupT	Tempo de escrita dos TCs ( $t_{ws}$ )	Float32	4
22	RSetupT	Tempo de leitura dos TCs ( $t_{rs}$ )	Float32	4
23	SPeriod	Período de recolha dos TCs ( $t_{sp}$ )	Float32	4
24	WarmUpT	Tempo de tolerância dos TCs	Float32	4
25	RDelayT	Tempo de atraso de leitura dos TCs ( $t_{ch}$ )	Float32	4
26	TestTime	Tempo necessário para executar um teste automático	Float32	4
Informação relacionada com o tempo de recolha				

27	TimeSrc	Recurso para o tempo de recolha	UInt8	1
28	InPropDI	Atraso na propagação da entrada dos dados	Float32	4
29	OutPropD	Atraso na propagação da saída dos dados	Float32	4
30	TSError	Tempo de atraso entre a execução do <i>trigger</i> e da recolha	Float32	4
Atributos				
31	Sampling	Atributos de recolha	—	—
48	SampMode	Capacidade do modo de recolha	UInt8	1
49	SDefault	Modo de recolha por predefinição	UInt8	1
32	DataXmit	Atributos da transmissão de informação	UInt8	1
33	Buffered	Atributos do <i>buffer</i>	UInt8	1
34	EndOfSet	Atributos da operação	UInt8	1
35	EdgeRpt	Atributos da informação da operação	UInt8	1
36	ActHalt	Atributos do atuador	UInt8	1
Sensibilidade				
37	Directon	Direção	Float32	4
38	DAngles	Ângulos de direção	Two Float32	8
Opções				
39	ESOption	Opções do sensor de eventos	UInt8	1
61-127	—	Reservado	—	—
128-255	—	Aberto aos fabricantes	—	—
—	—	<i>Checksum</i>	UInt16	2

Os campos obrigatórios do *TransducerChannel* TEDS são:

- **Tamanho:** é utilizado para definir o número de octetos que o TEDS descreve;
- **TEDSID:** contém a estrutura de dados definida no início de todos os TEDSs da norma IEEE 1451.0;
- **CalKey:** é responsável pela calibração do transdutor;
- **ChanType:** é responsável por definir qual é o tipo de transdutor que o TEDS vai descrever (sensor, atuador ou sensor de eventos);
- **PhyUnits:** é utilizado para descrever a unidade SI que o transdutor vai medir ou controlar. São permitidas as seguintes unidades: radiano (rad), esterradiano (sr), metro (m), quilograma (kg), segundo (s), ampere (A), kelvin (K), mole (mol) ou a candela (cd);
- **LowLimit:** é responsável por definir o valor mínimo que o transdutor vai medir ou controlar. Este valor é formado por 4 octetos e é representado em vírgula flutuante de acordo com a norma IEEE 754;



- **HiLimit:** é responsável por definir o valor máximo que o transdutor vai medir ou controlar. Este valor é formado por 4 octetos e é representado em vírgula flutuante de acordo com a norma IEEE 754;
- **OError:** é usado para descrever a variação de erro que ocorre na saída dos dados dos TCs;
- **SelfTest:** define a capacidade do transdutor realizar testes automáticos. O seu valor pode assumir dois valores:
  - 0 - No caso de não necessitar ou de não ser disponibilizado a função de testes automáticos;
  - 1 - No caso de a função ser disponibilizada.
- **Sample:** o campo *Sample* é composto por 3 subcampos obrigatórios:
  - **DatModel:** descreve o modelo de dados usado pelos TCs, quando este recebe um comando de leitura ou escrita;
  - **ModLenth:** define o número de octetos representados no campo *DatModel*.
  - **SigBits:** define o número de bits significativos do campo *DatModel*.
- **UpdateT:** define o máximo de tempo, em segundos, entre o iniciar de um evento *trigger* e a sua conclusão com a recolha da primeira amostragem;
- **WSetupT:** apenas é obrigatório para transdutores do tipo atuador. É necessário para definir o mínimo de tempo, em segundos, entre o momento em que a informação é escrita e o iniciar de um novo evento do tipo *trigger*;
- **RSetupT:** apenas é obrigatório para transdutores do tipo sensor. Este campo contém o máximo de tempo, em segundos, entre o momento em que o TIM recebe um comando do tipo *trigger* e o momento em que informação está disponível para ser lida;
- **SPeriod:** define o período mínimo de amostragem, expresso em segundos, dos TCs, tendo em conta, o tempo de conversão dos dados, a velocidade do processador e a complexidade do sistema;
- **WarmUpT:** contém o período de tempo, definido em segundos, para que os TCs estabilizem o seu desempenho para entrar novamente em modo de operação;
- **RDelayT:** define o máximo tempo de atraso, em segundos, entre o momento em que os TCs recebam um comando de leitura e o momento em que é iniciado a transmissão dos dados;
- **TestTime:** define o máximo de tempo, em segundos, para que os TCs realizem um teste automático;
- **Sampling:** é composto por 2 subcampos obrigatórios *SampMode* e *SDefault*. Estes dois atributos são utilizados para descrever qual o tipo de amostragem suportado pelos TCs;
- **EndOfSet:** apenas é obrigatório para transdutores do tipo atuador. Define os possíveis modos de operação que um atuador pode suportar;
- **EdgeRpt:** apenas é obrigatório para transdutores do tipo sensor de eventos. Define os possíveis modos de operação que um sensor de eventos pode suportar;
- **ActHalt:** apenas é obrigatório para transdutores do tipo atuador. Este valor determina o que um atuador deve fazer quando recebe um comando de paragem;

- **Checksum:** é o valor que garante a integridade do *TransducerChannel* TEDS.

Na tabela a.3 são apresentados os campos, obrigatórios ou opcionais, que definem o *User's Transducer Name* TEDS de acordo com a norma IEEE 1451.0 [22]. Além disso, é ainda descrito o identificador de cada campo, a sua descrição, e o seu respetivo tamanho.

**Tabela A.3: Estrutura do *User's Transducer Name* TEDS.**

ID do campo	Nome do Campo	Descrição	Tipo de Dados	N Octetos
—		Tamanho	UInt32	4
0-2	—	Reservado	—	—
3	TEDSID	TEDS <i>Identification Header</i>	UInt8	4
4	Format	Formato do TEDS	UInt8	1
5	TCName	Nome do TIM	—	—
—		<i>Checksum</i>	UInt16	2

Os campos obrigatórios do *User's Transducer Name* TEDS são:

- **Tamanho:** é utilizado para definir o número de octetos que o TEDS descreve;
- **TEDSID:** contém a estrutura de dados definida no início de todos os TEDSs da norma IEEE 1451.0;
- **Format:** define o formato do campo *TCName*. Este campo pode ser definido pelo utilizador ou pode ser baseado em texto;
- **TCName:** determina o nome do TIM, de acordo com a opção escolhida no campo *Format*;
- **Checksum:** é o valor que garante a integridade do *User's Transducer Name* TEDS.

Na tabela a.4 são apresentados os campos, obrigatórios ou opcionais, que definem o *End User Application Specific* TEDS de acordo com a norma IEEE 1451.0 [22]. Além disso, é ainda descrito o identificador de cada campo, a sua descrição, e o seu respetivo tamanho.

**Tabela A.4: Estrutura do *End User Application Specific* TEDS.**

ID do campo	Nome do Campo	Descrição	Tipo de Dados	N Octetos
—		Tamanho	UInt32	4
0-2	—	Reservado	—	—
3	TEDSID	TEDS <i>Identification Header</i>	UInt8	4
4-9	—	Reservado	—	—
10	EndUserData	Variável	—	Variável
—		<i>Checksum</i>	UInt16	2

Os campos obrigatórios do *End User Application Specific* TEDS são:

- **Tamanho:** é utilizado para definir o número de octetos que o TEDS descreve;
- **TEDSID:** contém a estrutura de dados definida no início de todos os TEDSs da norma IEEE 1451.0;
- **EndUserData:** permite que o utilizador defina um parâmetro, em qualquer formato, para ser guardado no TIM;
- **Checksum:** é o valor que garante a integridade do *End User Application Specific* TEDS.



# Anexo B

## Interfaces Web

Este anexo apresenta as várias interfaces *Web* das ferramentas disponíveis na aplicação desenvolvida. Estas referem-se às ferramentas: i) *Meta-TEDS*, ii) *TransducerChannel TEDS*, iii) *User's Transducer Name TEDS*, iv) *End User Application Specific TEDS*, v) *Manufacturer-defined TEDS*, vi) *Config Map Table* e, vii) *Configure Weblab Modules Connections*.

The screenshot displays the Meta-TEDS web interface. At the top, it has a title 'Meta-TEDS' and a breadcrumb 'Home > Meta-TEDS'. Below this is a quote: "THE META-TEDS IS A REQUIRED TEDS. THE FUNCTION OF THE META-TEDS SHALL BE TO MAKE AVAILABLE AT THE INTERFACE ALL INFORMATION NEEDED TO GAIN ACCESS TO ANY TRANSDUCERCHANNEL, PLUS INFORMATION COMMON TO ALL TRANSDUCERCHANNELS." followed by "...in IEEE Std 1451.0™-2007".

The main section is titled 'Read TEDS From File' and contains a 'Search File:' input field with a 'SUBMIT' button. Below this are 'Length (Dec)' and '(Hexa)' input fields. The current time is '11:31:46 PM' and the date is '2013-11-13'.

The 'Meta-TEDS Identifier' section includes fields for 'Type' (03), 'Length' (04), 'Family' (00), 'Class' (01), 'MetaTEDS', 'Version' (01), and 'Tuple Length' (01).

The 'Universal Unique Identification Data' section contains fields for 'Address', 'County', 'PostCode', 'Manufacturer', 'Town/City', 'Latitude', 'Longitude', 'Date' (2013-11-13), 'Country' (Portugal), and 'Sequence'. A note states: "(Note: Latitude and Longitude are calculated automatically. Please complete all fields.)".

The 'Timing-Related Information' section has fields for 'Operational Time-Out (sec.)', 'Slow-Access Time-Out (sec.)', and 'Self-Test Time (sec.)'.

The 'TransducerChannels' section has a 'Number of Implemented TransducerChannels' input field and a 'Create Channels' button.

The 'Open To Manufacturers' section features a horizontal bar with 'Add New Tuple/Length/Value', a '+' sign, a '-' sign, and 'Remove Last Tuple/Length/Value'.

At the bottom, there are 'Checksum (Dec)' and '(Hexa)' input fields. A 'Build TEDS' section contains 'Build TEDS', 'Create TEDS', and 'Save TEDS' buttons. A 'BUILD' button is highlighted with a tooltip: "THIS BUTTON IS USED TO VALIDATE ALL FORM FIELDS." The breadcrumb 'Home > Meta-TEDS' is repeated at the very bottom.

Figura B.1: Interface da ferramenta *Meta-TEDS*.

**TransducerChannel TEDS**  
[Home](#) » TransducerChannel TEDS

*"THIS IS A REQUIRED TEDS. THE FUNCTION OF THE TRANSDUCERCHANNEL TEDS SHALL BE TO MAKE AVAILABLE AT THE INTERFACE ALL OF THE INFORMATION CONCERNING THE TRANSDUCERCHANNEL BEING ADDRESSED TO ENABLE THE PROPER OPERATION OF THE TRANSDUCERCHANNEL.."*  
 ...in IEEE Std 1451.0™-2007

**Read TEDS From File**

Search File:

Length (Dec)  (Hexa)  Time: 11:37:49 PM Date: 2013-11-13

**TransducerChannel TEDS Identifier**

Type  Length  Family  Class  ChanTEDS Version  Tuple Length

**TransducerChannel Related Information**

Calibration Key  CAL\_NONE TransducerChannel Type Key  Sensor

Physical Units Interpretation  PUI\_SI\_UNITS

Radians (rad)  Steradians (sr)  Meters (m)  Kilograms (kg)  Seconds (s)

Amperes (A)  Kelvins (K)  Moles (mol)  Candelas (cd)  Units Extension  <sup>op</sup>

Operational Lower Range Limit (SI)  Operational Upper Range Limit (SI)  Worst-Case Uncertainty

Self-Test Key  No self-test provided Multi-Range Capability  <sup>op</sup>

Data Model  N-Octet Integer Data Model Length  Model Significant Bits

*(Note: Data Model Length is calculated automatically when you validate all form fields. Please complete Model Significant Bits field.)*

Maximum Data Repetitions  <sup>op</sup> Series Origin  Series Increment  Maximum Pre-trigger

Series Units Interpretation  PUI\_SI\_UNITS Series Units Type (SI)  Series Units Value

*(Note: Series Origin, Series Increment, Series Units and Maximum Pre-trigger depends on the Maximum Repetitions field.)*

**Timing-Related Information**

Update Time (s)  Write Setup Time (s)  Read Setup Time (s)  Sampling Period (s)

Warm-Up Time (s)  Read Delay Time (s)  Self-Test Time Requirement (s)

**Timing-Sample Information**

Sample Time  NoHelp <sup>op</sup> Incoming Delay (s)  Outgoing Delay (s)  Trigger Delay (s)  <sup>op</sup>

**Attributes**

Sampling Mode Capability  Trigger Initiated Mode Default Sampling Mode

Buffered Attribute  <sup>op</sup> End-Of-Data-Set Operation  Data Transmission  <sup>op</sup> Edge-To Report  Actuator-Halt

**Sensitivity**

Sensitivity Direction  NO <sup>op</sup> Direction Angles ( $\rho$  / rad)  <sup>op</sup> ( $\Phi$  / rad)  <sup>op</sup>

**Options**

Event Sensor Options  Not Applicable

**Open To Manufacturers**

Add New Tuple/Length/Value   Remove Last Tuple/Length/Value

Checksum (Dec)  (Hexa)

THIS BUTTON IS USED TO VALIDATE ALL FORM FIELDS.

[Home](#) » TransducerChannel TEDS

**Figura B.2: Interface da ferramenta TransducerChannel TEDS.**

### User's Transducer Name TEDS

[Home](#) » User's Transducer Name TEDS

*"THIS FIELD IS A REQUIRED TEDS FOR THE TIM AND IS RECOMMENDED FOR ALL TRANSDUCERS. THE USER'S TRANSDUCER NAME TEDS PROVIDES A PLACE TO STORE THE NAME BY WHICH THE SYSTEM OR THE END USER WILL KNOW THIS TRANSDUCER."*

...in IEEE Std 1451.0™-2007

**Read TEDS From File**

Search File:

Length (Dec)  (Hexa)  Time: 6:28:20 PM Date: 2014-03-22

**User's Transducer Name TEDS Identifier**

Type  Length  Family  Class   Version  Tuple Length

**Format Description Of This TEDS**

Format  Value  Meaning

**TIM Or TransducerChannel Name**

TIM or TransducerChannel Name

Checksum (Dec)  (Hexa)

---

THIS BUTTON IS USED TO VALIDATE ALL FORM FIELDS.

---

[Home](#) » User's Transducer Name TEDS

**Figura B.3: Interface da ferramenta *User's Transducer Name TEDS*.**

### End User Application Specific TEDS

[Home](#) » End User Application Specific TEDS

*"THIS FIELD IS AN OPTIONAL TEDS THAT PROVIDES STORAGE FOR APPLICATION-DEPENDENT DATA THAT THE USER WANTS TO KEEP WITH THE TIM OR TRANSDUCERCHANNEL. THE USER SHALL DETERMINE THE CONTENT AND FUNCTION OF THE END USER APPLICATION SPECIFIC TEDS."*

...in IEEE Std 1451.0™-2007

**Read TEDS From File**

Search File:

Length (Dec)  (Hexa)  Time: 12:10:22 AM Date: 2013-11-14

**Manufacturer-Defined TEDS Identifier**

Type  Length  Family  Class   Version  Tuple Length

**End User Data**

Data Type  Octets  End User Data

Checksum (Dec)  (Hexa)

---

THIS BUTTON IS USED TO VALIDATE ALL FORM FIELDS.

---

[Home](#) » End User Application Specific TEDS

**Figura B.4: Interface da ferramenta *End User Application Specific TEDS*.**

**Manufacturer-Defined TEDS**

[Home](#) » [Manufacturer-Defined TEDS](#)

\*MANUFACTURER-DEFINED TEDS MAY BE IN ANY FORMAT REQUIRED BY THE MANUFACTURER'S APPLICATION SOFTWARE. FOR A MANUFACTURER-DEFINED TEDS THAT IS BEING SENT TO THE TIM, THE SYSTEM SHALL TAKE THE INFORMATION, APPLY THE LENGTH FIELD AND CHECKSUM FIELDS, AND TRANSMIT IT TO THE TIM. THIS IS A OPTIONAL TEDS.\*

...in IEEE Std 1451.0™-2007

**Read TEDS From File**

Search File:

Length (Dec)  (Hexa)  Time: 12:05:26 AM Date: 2013-11-14

**Manufacturer-Defined TEDS Identifier**

Type  Length  Family  Class  Version  Tuple Length

**Free Data Block Contents**

Add New Tuple/Length/Value   Remove Last Tuple/Length/Value

---

Checksum (Dec)  (Hexa)

---

THIS BUTTON IS USED TO VALIDATE ALL FORM FIELDS.

---

[Home](#) » [Manufacturer-Defined TEDS](#)

**Figura B.5: Interface da ferramenta *Manufacturer-defined TEDS*.**

**Config Map Table**

[Home](#) » [Config Map Table](#)

\*MAP TABLE HAS A STRUCTURE SIMILAR TO A TEDS. USED TO ASSOCIATE TEDS ID TO A MEMORY NUMBER THAT WILL BE HANDLED BY THE IEEE 1451.0-MODULE.\*

...new approach for WebLab Configuration

**Read Map Table From File**

Search File:

Length (Dec)  (Hexa)  Time: 12:13:20 AM Date: 2013-11-14

**Transducer Interface Module**

Transducer Interface Module ID

Add New TEDS   Remove Last TEDS

---

**Transducer Channels Group**

Add New Transducer Channel   Remove Last Transducer Channel

**Transducer Channels**

Transducer Channels 1 ID

Add New TEDS   Remove Last TEDS

---

Checksum (Dec)  (Hexa)

---

THIS BUTTON IS USED TO VALIDATE ALL FORM FIELDS.

---

[Home](#) » [Config Map Table](#)

**Figura B.6: Interface da ferramenta *Config Map Table*.**



## Configure Weblab Modules Connections

[Home](#) » [Configure Weblab Modules Connections](#)

*"CONFIGURATION FILE USED TO: I) CHECK TEDS AND MAP TABLE CONFIGURATIONS; II) GENERATE VERILOG TEDS AND MAP TABLE FILES; III) INTERFACE TEDS FILES AND MAP TABLE TO THE IEEE1451-INFRASTRUCTURE."*

### Read Configuration WebLab File

Search File:

Time: 10:06:18 PM    Date: 2014-10-25

**IEEE1451.0-Compliant Module**

Module Description

Add TEDS      Remove TEDS

---

TEDS File  Attribute  Status  Description

---

Mask Module Register Name  Mask Module Register (Hexa)

**Weblab Modules**

Add Weblab Modules      Remove Weblab Modules

**Weblab Modules 1**

Main WebLab HDL Module Name

Add Tranducer Channel      Remove Tranducer Channel

**Tranducer Channel 1**

Add TEDS      Remove TEDS

---

TEDS File  Attribute  Status  Description

---

Mask Register Name  Mask Register (Hexa)  Task Location (.vh)

---

Select the Associated Tasks:

Start  Stop  Read  Write  Init  Update  Event

---

Add Wire      Remove Wire

---

Name  Event  Width  Direction

---

Add Variable      Remove Variable

---

Variable Name  Value

**External Wires**

(Note: The UCF file must be in accordance with these wires.)

Add External Wire      Remove External Wire

---

Name  Clk  Width  Direction

---

Event

**HDL Verilog Files**

Add File      Remove File

---

Weblab Module File Location (.v)

**General Definitions**

Map Table File  Memory Buffer  Clk Freq. (MHz)  <sup>op.</sup> Bits per second (s)  <sup>op.</sup> Baud Rate

*THIS BUTTON IS USED TO VALIDATE ALL FORM FIELDS.*

[Home](#) » [Configure Weblab Modules Connections](#)

**Figura B.7: Interface da ferramenta *Configure Weblab Modules Connections*.**

101



# Anexo C

## Modelo relacional da base de dados

Este anexo apresenta o modelo relacional da base de dados da aplicação, como é ilustrado na figura C.1. Este modelo, criado com base na norma IEEE 145.0, é composto por vinte e sete tabelas, nomeadamente:

- “teds\_attributes”: tabela com a informação associada aos atributos dos TEDSs;
- “teds\_identifierheader”: tabela com a informação da identificação dos TEDSs;
- “teds\_versionenumeration”; tabela com a informação associada às versões dos TEDSs;
- “teds\_physicalunits”: tabela com a informação necessária para a interpretação das unidades físicas;
- “teds\_classenumeration”: tabela com a descrição de todas as classes dos TEDSs;
- “teds\_metateds”: tabela necessária para definir o *META*-TEDS. É uma das tabelas principais do modelo relacional;
- “teds\_tc\_controlgroup”: tabela com a informação associada ao grupo de controlo (*Meta*-TEDS);
- “teds\_tc\_controlgroup\_type”: tabela com a lista de membros do grupo controlo (*Meta*-TEDS);
- “teds\_tc\_vectorgroup”: tabela com a informação associada ao grupo de vetor (*Meta*-TEDS);
- “teds\_tc\_vectorgroup\_type”: tabela com a lista de membros do grupo vetor (*Meta*-TEDS);
- “teds\_tc\_geogroup”: tabela com a informação associada ao grupo de localização geográfica (*Meta*-TEDS);
- “teds\_tc\_geogroup\_type”: tabela com a lista de membros do grupo vetor (*Meta*-TEDS);
- “teds\_tc\_geoenumeration”: tabela com a informação do tipo de localização (*Meta*-TEDS);
- “teds\_tc\_transducergroup”: tabela com a informação da organização dos TCs *proxy* (*Meta*-TEDS);

- “teds\_transducerchannel”: tabela necessária para definir o *TransducerChannel* TEDS. É uma das tabelas principais do modelo relacional;
- “teds\_calibrationkeys”: tabela com a informação da calibração do transdutor (*TransducerChannel* TEDS);
- “teds\_transducerchannelkeys”: tabela com a informação do tipo de transdutor (*TransducerChannel* TEDS);
- “teds\_selftestkey”: tabela com a informação relacionada com os testes automáticos (*TransducerChannel* TEDS);
- “teds\_datamodelsenumeration”: tabela com a informação associada ao modelo de dados utilizado (*TransducerChannel* TEDS);
- “teds\_timesampleinformation”; tabela com a informação do tempo de recolha dos dados (*TransducerChannel* TEDS);
- “teds\_samplingmode”: tabela com a informação associada aos modos de recolha dos dados (*TransducerChannel* TEDS);
- “teds\_sensitivityenumeration”: tabela com a informação com o tipo de direção (*TransducerChannel* TEDS);
- “teds\_sensoroption”: tabela com a informação associada ao sensor de eventos (*TransducerChannel* TEDS);
- “teds\_enduserapplication”: tabela necessária para definir o *End User Application Specific* TEDS. É uma das tabelas principais do modelo relacional;
- “teds\_usertransducer”: tabela necessária para definir o *User’s Transducer Name* TEDS. É uma das tabelas principais do modelo relacional;
- “teds\_formatenumeration”: tabela com a informação relacionada com o formato do nome do transdutor (*User’s Transducer Name* TEDS);
- “teds\_manufacturer”: tabela necessária para definir o *Manufacturer-defined* TEDS. É uma das tabelas principais do modelo relacional.

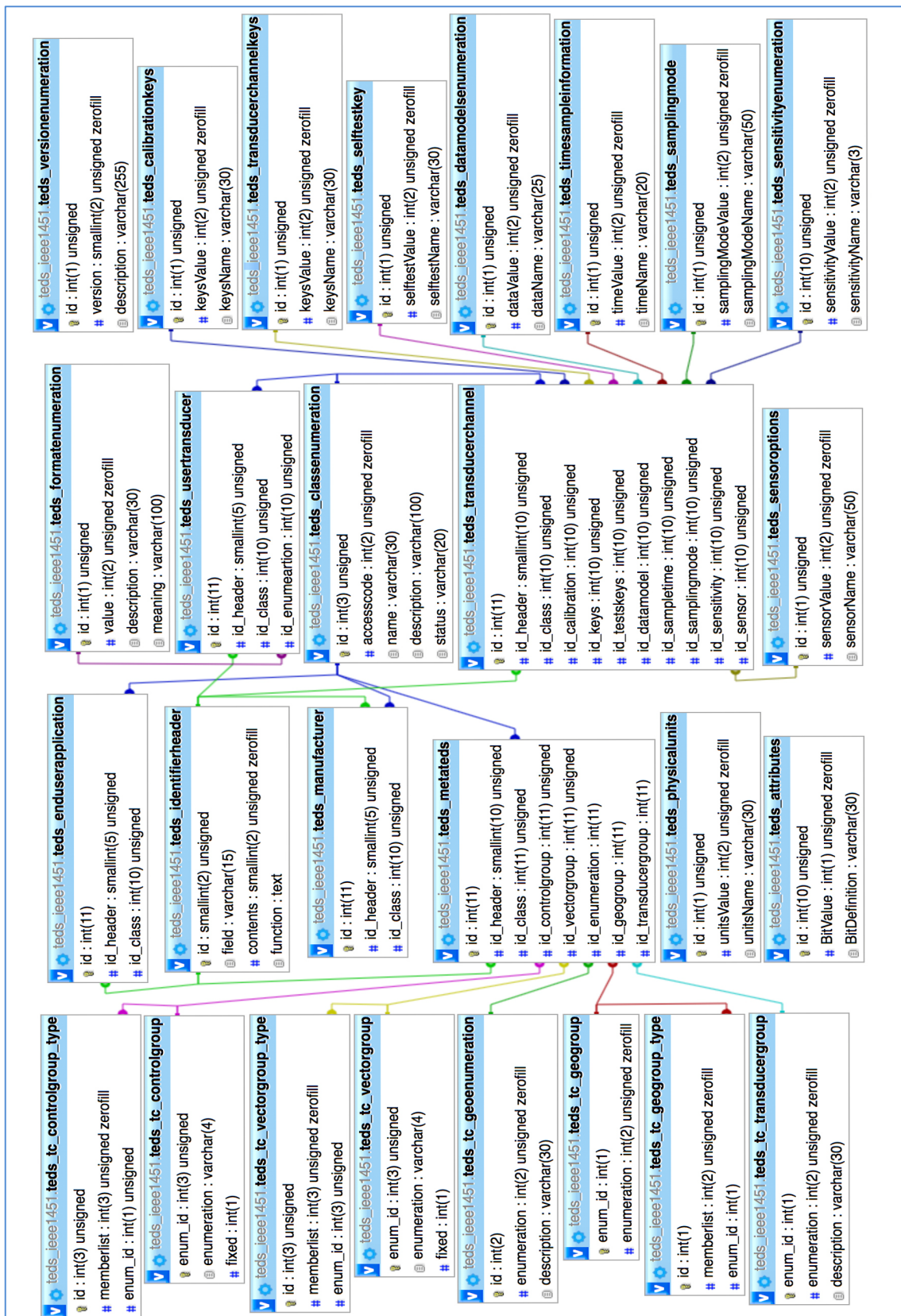


Figura C.1: Modelo relacional da base de dados da aplicação desenvolvida.



# Anexo D

## Ficheiros XML de validação

Este anexo apresenta os ficheiros XML utilizados para validar cada uma das ferramentas disponíveis na aplicação desenvolvida. Estes ficheiros XML referem-se às ferramentas: i) *Meta-TEDS*, ii) *TransducerChannel TEDS*, iii) *User's Transducer Name TEDS*, iv) *End User Application Specific TEDS*, v) *Manufacturer-defined TEDS* e, vi) *Config Map Table*.

### Ficheiro XML de validação do *Meta-TEDS*:

```
<transducer>
<teds name="MetaTEDS">
  <field name="TEDSID" description="TEDS Identification Header">
    <element name="Type" value="03" type="integer" matter="required" max="255"
      explain=" always has the value '03' in Meta-TEDS structure;" />
    <element name="Value" value="04" />
    <element name="Length" value="04" type="integer" matter="required" max="255"
      explain=" field always contains '04' octets;" />
    <element name="Family" value="00" type="integer" matter="required" max="255"
      explain=" field is used to identifies the member os the IEEE Std 1451 family. The value is '0';" />
    <element name="Class" value="01" type="integer" matter="required" max="255"
      explain=" always has the value '01' to identifies Meta-TEDS access code;" />
    <element name="Version" value="01" type="integer" matter="required" max="255"
      explain=" always has the value '01', indicating initial release of this standard;" />
    <element name="TupleLength" value="01" type="integer" matter="required" max="255"
      explain=" has the value '01', meaning that there are 255 or less octets in the value field;" />
  </field>
  <field name="UUID" description="Globally Unique Identifier">
    <element name="Type" value="04" />
    <element name="Length" value="10" />
    <element name="N" size="1" />
    <element name="S" size="0" />
    <element name="E" size="1" />
    <element name="W" size="0" />
    <element name="Latitude" size="20" type="string" matter="required"/>
    <element name="Longitude" size="20" type="string" matter="required"/>
    <element name="Manufacturer" size="4" type="integer" matter="required" max="15"
      explain=" field has a value between '0' and '15';" />
    <element name="Year" size="12" type="string" matter="required" />
    <element name="Sequence" size="22" type="integer" matter="required" min="1" max="255"
      explain=" field has a value between '1' and '255';" />
  </field>
  <field name="OHoldOff" description="Operational time-out">
    <element name="Type" value="10" />
    <element name="Length" value="04" />
    <element name="Operational" type="float" matter="required" max="65535"
      explain=" field has a value up '65535';" />
  </field>
  <field name="SHoldOff" description="Slow-access time-out">
    <element name="Type" value="11" />
    <element name="Length" value="04" />
    <element name="SlowAccess" type="float" matter="optional" max="65535"
      explain=" field has a value up '65535';" />
  </field>
</teds>
</transducer>
```

```

<field name="TestTime" description="Self-Test Time">
  <element name="Type" value="12" />
  <element name="Length" value="04" />
  <element name="SelfTest" type="float" matter="required" max="65535"
    explain=" field has a value up '65535';"/>
</field>
<field name="MaxChan" description="Number of implemented TransducerChannels">
  <element name="Type" value="13" />
  <element name="Length" value="02" />
  <element name="NChannel" type="integer" matter="required" min="1" max="255"
    explain= " field has a value between '1' and '255';"/>
</field>
<field name="CGroup" description="ControlGroup information sub-block">
  <element name="Type" value="14" />
  <element name="Length" value="" />
  <element name="GrpType" value="20" size="01" />
  <element name="MemList" value="21" type="integer" matter="required" min="0" max="255"
    explain= " field has a value between '0' and '255';"/>
</field>
<field name="VGroup" description="VectorGroup information sub-block">
  <element name="Type" value="15" />
  <element name="Length" value="" />
  <element name="GrpType" value="20" size="01" />
  <element name="MemList" value="21" type="integer" matter="required" min="0" max="255"
    explain= " field has a value between '0' and '255';"/>
</field>
<field name="GeoLoc" description="Specialized VectorGroup for geographic location">
  <element name="Type" value="16" />
  <element name="Length" value="" />
  <element name="LocEnum" value="24" size="01" />
  <element name="GrpType" value="20" size="01" />
  <element name="MemList" value="21" type="integer" matter="required" min="0" max="255"
    explain= " field has a value between '0' and '255';"/>
</field>
<field name="Proxies" description="TransducerChannel Proxy definition sub-block">
  <element name="Type" value="17" />
  <element name="Length" value="" />
  <element name="ChanNum" value="22" type="integer" matter="required" min="1" max="255"
    explain= " field has a value between '1' and '255';"/>
  <element name="Organiz" value="23" size="01" />
  <element name="MemList" value="21" type="integer" matter="required" min="1" max="255"
    explain= " field has a value between '1' and '255';"/>
</field>
</teds>
</transducer>

```

### Ficheiro XML de validação do *TransducerChannel* TEDS:

```

<transducer>
<teds name="TransducerChannelTEDS">
  <field name="TEDSID" description="TEDS Identification Header">
    <element name="Type" value="03" type="integer" matter="required" max="255"
      explain=" always has the value '03' in TransducerChannelTEDS structure;"/>
    <element name="Value" value="04" />
    <element name="Length" value="04" type="integer" matter="required" max="255"
      explain=" field always contains '04' octets;"/>
    <element name="Family" value="00" type="integer" matter="required" max="255"
      explain=" field is used to identifies the member os the IEEE Std 1451 family. The value is '0';"/>
    <element name="Class" value="03" type="integer" matter="required"
      explain=" always has the value '03' to identifies TransducerChannelTEDS access code;"/>
    <element name="Version" value="01" type="integer" matter="required" max="255"

```



```

        explain=" always has the value '01', indicating initial release of this standard;"/>
        <element name="TupleLength" value="01" type="integer" matter="required" max="255"
        explain=" has the value '01', meaning that there are 255 or less octets in the value field;"/>
    </field>
    <field name="RELATED" description="TEDS Related Information">
        <element name="UnitType" type="integer" matter="required" min="0" max="255"
        explain=" field has a value between '0' and '255';"/>
        <element name="Operational" type="float" matter="required"
        explain=" Operational Lower Range Limit should be less than Operational Upper Range Limit;"
        explainAux=" Operational Upper Range Limit must be greater than the Operational Lower
Range Limit;"/>
        <element name="MRange" type="integer" min="0" max="255"
        explain=" field has a value between '0' and '255';"/>
        <element name="DataSet" type="integer" min="0" max="65535"
        explain=" field has a value between '0' and '65535';"/>
    </field>
    <field name="TIMMING" description="TEDS Timming Related Information">
        <element name="Time" type="integer" matter="required" />
    </field>
    <field name="TIMESAMPLE" description="TEDS Time of the Sample Information">
        <element name="Sample" type="integer" />
        <element name="Trigger" type="integer" matter="optional" />
    </field>
    <field name="SENSITIVITY" description="TEDS Sensitivity">
        <element name="Angles" type="integer" matter="optional" />
    </field>
</teds>
</transducer>

```

#### **Ficheiro XML de validação do *User's Transducer Name TEDS*:**

```

<transducer>
<teds name="UserTransducerNameTEDS">
    <field name="TEDSID" description="TEDS Identification Header">
        <element name="Type" value="03" type="integer" matter="required" max="255"
        explain=" always has the value '03' in User's Transducer Name TEDS structure;"/>
        <element name="Value" value="04" />
        <element name="Length" value="04" type="integer" matter="required" max="255"
        explain=" field always contains '04' octets;"/>
        <element name="Family" value="00" type="integer" matter="required" max="255"
        explain=" field is used to identifies the member os the IEEE Std 1451 family. The value is '0';"/>
        <element name="Class" value="12" type="integer" matter="required" max="255"
        explain=" always has the value '12' to identifies User's Transducer Name TEDS access code;"/>
        <element name="Version" value="01" type="integer" matter="required" max="255"
        explain=" always has the value '01', indicating initial release of this standard;"/>
        <element name="TupleLength" value="01" type="integer" matter="required" max="255"
        explain=" has the value '01', meaning that there are 255 or less octets in the value field;"/>
    </field>
    <field name="Format" description="Format description of this TEDS">
        <element name="FormatField" value="01"
        explain=" The structure of Text-based TEDS are not defined in this application;"/>
    </field>
    <field name="TCName" description="Tim or TransducerChannel Name">
        <element name="TCNameField" value="05" matter="required"/>
    </field>
</teds>
</transducer>

```

#### **Ficheiro XML de validação do *End User Application Specific TEDS*:**

```

<transducer>
<teds name="EndUserApplicationSpecificTEDS">

```

```

<field name="TEDSID" description="TEDS Identification Header">
  <element name="Type" value="03" type="integer" matter="required" max="255"
    explain=" always has the value '03' in End User Application Specific TEDS structure;" />
  <element name="Value" value="04" />
  <element name="Length" value="04" type="integer" matter="required" max="255"
    explain=" field always contains '04' octets;" />
  <element name="Family" value="00" type="integer" matter="required" max="255"
    explain=" field is used to identifies the member os the IEEE Std 1451 family. The value is '0';" />
  <element name="Class" value="07" type="integer" matter="required" max="255"
    explain=" always has the value '07' to identifies End User Application Specific TEDS access
code;" />
  <element name="Version" value="01" type="integer" matter="required" max="255"
    explain=" always has the value '01', indicating initial release of this standard;" />
  <element name="TupleLength" value="01" type="integer" matter="required" max="255"
    explain=" has the value '01', meaning that there are 255 or less octets in the value field;" />
</field>
</teds>
</transducer>

```

### Ficheiro XML de validação do *Manufacturer-defined TEDS*:

```

<transducer>
<teds name="ManufacturerDefinedTEDS">
  <field name="TEDSID" description="TEDS Identification Header">
    <element name="Type" value="03" type="integer" matter="required" max="255"
      explain=" always has the value '03' in Manufacturer TEDS structure;" />
    <element name="Value" value="04" />
    <element name="Length" value="04" type="integer" matter="required" max="255"
      explain=" field always contains '04' octets;" />
    <element name="Family" value="00" type="integer" matter="required" max="255"
      explain=" field is used to identifies the member os the IEEE Std 1451 family. The value is '0';" />
    <element name="Class" type="integer" matter="required" min="128" max="255"
      explain=" field has a value between '128' and '255';" />
    <element name="Version" value="01" type="integer" matter="required" max="255"
      explain=" always has the value '01', indicating initial release of this standard;" />
    <element name="TupleLength" value="01" type="integer" matter="required" max="255"
      explain=" has the value '01', meaning that there are 255 or less octets in the value field;" />
  </field>
  <field name="FreeDataBlock" description="Format description of this TEDS">
    <element name="TypeField" type="integer" matter="required" min="4" max="255"
      explain=" Field has a value between '4' and '255';" />
  </field>
  <field name="FreeDataBlockMetaTEDS" description="Format description of this TEDS">
    <element name="TypeField" type="integer" matter="required" min="128" max="255"
      explain=" Field has a value between '128' and '255';" />
  </field>
</teds>
</transducer>

```

### Ficheiro XML de validação do *Config Map Table*:

```

<transducer>
<teds name="MapTable">
  <field name="MAP" description="Map Table Config">
    <element name="TIM" type="integer" matter="required" min="128" max="255"
      explain=" field has a value between '128' and '255';" />
    <element name="TC" type="integer" matter="required" min="128" max="255"
      explain=" field has a value between '128' and '255';" />
  </field>
</teds>
</transducer>

```