



Instituto Superior de
Engenharia de Coimbra

Departamento
de Engenharia Informática e de Sistemas

Integração do iMod

Dissertação apresentada para a obtenção do grau de Mestre em Informática e Sistemas, com especialização em Desenvolvimento de *Software*

Autor

Kévin Pascoal Cadima

Orientador ISEC

Viriato António Pereira Marinho Marques

Instituto Politécnico de Coimbra, Instituto Superior de Engenharia de
Coimbra

Orientadora ISA

Rita Patrícia Dinis Carreira

ISA – Intelligent Sensing Anywhere

Coimbra, novembro, 2014

AGRADECIMENTOS

A todos aqueles que me acompanharam durante o desenvolvimento do presente projeto e acreditaram nas minhas capacidades para a sua concretização, expresse a minha sincera gratidão.

À ISA – Intelligent Sensing Anywhere e, particularmente, à Eng. Andreia Carreiro, ao Eng. David Macário e ao Eng. Pedro Mêdas por me terem confiado este projeto.

Ao Prof. Doutor Viriato Marques, por toda a orientação e disponibilidade que manifestou ao longo deste percurso.

À Eng. Rita Carreira por toda a orientação, dedicação, empenho e conhecimento com que direcionou e acompanhou o meu projeto ao fim de poder ser concretizado.

Finalmente, de modo especial, quero agradecer à minha família e amigos que me apoiaram, acreditando sempre em mim.

Cordialmente.

RESUMO

O presente relatório descreve o projeto de estágio que decorreu na empresa ISA – Intelligent Sensing Anywhere – e consistiu na integração nas soluções ISA dum módulo da empresa Techbase denominado iMod.

Para efeitos de controlo com vista ao aumento da eficiência energética, uma instituição bancária decidiu optar por dispositivos de recolha e controlo nas suas várias agências, a fim de reduzir a energia consumida nas respetivas instalações, e por consequência poder reduzir custos. Numa era em que a ecologia é importante, é de realçar também o papel deste investimento na preservação do ambiente.

A instituição bancária em questão recorreu ao *know-how* da ISA para o desenvolvimento da solução pretendida. O dispositivo de recolha e controlo nas agências bancárias é designado por iMod.

PALAVRAS-CHAVE

Eficiência Energética

Gestão de Energia

iMod

ISA- Intelligent Sensing Anywhere

Linguagem C

Linux

PHP

SCRUM

Techbase

ABSTRACT

This report describes the internship's project which took place in the company ISA - Intelligent Sensing Anywhere - and it consisted in the integration in the ISA solutions of a module device from the company Techbase named iMod.

For control purposes aiming to increase energy efficiency, a bank decided to opt for collection and control devices in their various agencies in order to reduce the energy consumed in the respective facilities, and consequently reduce costs. In an era in which ecology is important, it is also noted the role of this investment in preserving the environment.

The bank in question appealed to the know-how of ISA for the development of the desired solution. The collection and control device in the banks is called iMod.

KEYWORDS

C language

Energy Efficiency

Energy management

iMod

ISA- Intelligent Sensing Anywhere

Linux

PHP

SCRUM

Techbase

ÍNDICE

1. INTRODUÇÃO	16
1.1 ÂMBITO	16
1.2 APRESENTAÇÃO DA EMPRESA.....	16
1.3 OBJETIVOS DO ESTÁGIO.....	17
1.4 ORGANIZAÇÃO E TEMAS ABORDADOS NO PRESENTE RELATÓRIO.....	18
2. ANÁLISE DO PROBLEMA	19
2.1 ÂMBITO INICIAL	19
2.1.1 O iMod	19
2.2 INVESTIGAÇÃO DO IMOD E TECNOLOGIAS.....	20
2.3 DESENVOLVIMENTO INTEGRADO EM PROJETO PARA CLIENTE	20
2.3.1 Visão e âmbito do projeto	21
2.3.2 Principais entidades do sistema.....	21
2.3.3 Descrição Geral do Projeto de Estágio.....	23
3. PLANO DE TRABALHOS DO ESTÁGIO	24
3.1 METODOLOGIA DE DESENVOLVIMENTO	24
3.1.1 Descrição do Scrum.....	24
3.2 APLICAÇÃO DO SCRUM NO PROJETO DE ESTÁGIO.....	25
3.2.1 <i>Pre-Game</i> (Análise)	25
3.2.2 <i>Game</i> (Análise, Desenho, Construção, Pré-produção).....	27
3.2.3 <i>Post-Game</i> (Operacionalização)	28
4. FASE DE INTEGRAÇÃO E INVESTIGAÇÃO.....	29
4.1 CONTEXTUALIZAÇÃO	29
4.2 SCADA	31
4.3 IMOD	33
4.4 BIBLIOTECA MODBUS	35
4.4.1 Modbus.....	36
4.4.2 Desenvolvimento experimental – cliente Modbus-TCP (Mestre) – servidor TCP.....	36
4.4.3 Criação de documento.....	36
4.4.4 Desenvolvimento experimental – cliente Modbus-TCP (Mestre) – servidor Modbus-TCP (Escravo)	37

4.4.5	Desenvolvimento experimental – cliente Modbus-TCP (Escravo) – servidor Modbus-TCP (Mestre)	39
4.5	COMUNICAÇÃO POR PORTA SÉRIE	39
4.5.1	GPRS e pesquisa de biblioteca Java que implemente comunicação via GPRS	39
4.5.2	Comandos AT.....	39
4.5.3	GSM	40
4.6	Programas na linguagem C no iMod	41
4.7	RESULTADOS OBTIDOS NESTA FASE	41
5.	FASE DE REQUISITOS	43
5.1	VISÃO GERAL.....	43
5.2	CARACTERÍSTICAS DOS UTILIZADORES	44
5.3	ANÁLISE DE REQUISITOS	44
5.3.1	Metodologia Usada na Recolha de Requisitos.....	44
5.4	REQUISITOS FUNCIONAIS.....	45
5.4.1	Interface Web do iMod	45
5.5	PROTÓTIPOS DE INTERFACE.....	46
5.6	REQUISITOS NÃO-FUNCIONAIS.....	46
5.6.1	Eficiência.....	46
5.6.2	Fiabilidade.....	46
5.6.3	Manutenção	47
5.6.4	Portabilidade.....	47
5.6.5	Segurança	47
5.6.6	Usabilidade.....	47
5.6.7	Requisitos Tecnológicos	47
5.7	ANÁLISE CRÍTICA	47
6.	ANÁLISE TECNOLÓGICA.....	48
6.1	ESTADO DO PROJETO	48
6.2	ANÁLISE DE TECNOLOGIAS PARA O SISTEMA.....	48
6.3	FERRAMENTAS DE DESENVOLVIMENTO DE SOLUÇÕES EM PHP	48
6.4	PHP NO IMOD	49
6.5	APRENDIZAGEM DE PHP	49
6.6	CONCLUSÕES.....	49

7. FASE DE ARQUITETURA E IMPLEMENTAÇÃO	50
7.1 DEFINIÇÃO DE ARQUITETURA GERAL.....	50
7.2 DEFINIÇÃO DE ARQUITETURA EM ÂMBITO DE IMPLEMENTAÇÃO.....	50
7.3 IMPLEMENTAÇÃO.....	51
7.3.1 Ferramentas.....	54
7.3.2 Organização do Código.....	54
7.4 COMPONENTES.....	55
7.4.1 Interface Web do iMod.....	55
7.4.2 API do iHub.....	56
7.5 DESIGN DE INTERFACE.....	58
7.6 ANÁLISE CRÍTICA.....	58
8. FASE DE VERIFICAÇÃO E VALIDAÇÃO	59
8.1 PLANO DE TESTES.....	59
8.1.1 Elaboração do plano de testes.....	59
8.1.2 Execução do plano de testes.....	59
8.2 REVISÃO.....	61
8.3 VALIDAÇÃO.....	61
8.4 VERIFICAÇÃO.....	62
8.5 INSTALAÇÃO NO CLIENTE.....	62
9. CONCLUSÃO E PERSPETIVAS FUTURAS	63
9.1 PLANEAMENTO EFECTIVAMENTE SEGUIDO.....	63
9.1.1 Justificação dos Desvios.....	63
9.2 AVALIAÇÃO DO TRABALHO DESENVOLVIDO.....	63
9.3 TRABALHO FUTURO.....	63
9.4 CONSIDERAÇÕES PESSOAIS.....	63
10. REFERÊNCIAS BIBLIOGRÁFICAS	65
11. ANEXOS	68
A. Anexo A - Proposta de Estágio.....	68
B. Anexo B - Procura de Biblioteca Modbus-TCP.....	71
1 Introdução	73
2. Biblioteca para Modbus TCP	74

2.1	Pesquisa de biblioteca Modbus TCP	74
2.2	Apresentação genérica de biblioteca encontrada	74
2	Implementação de comunicação usando Modbus TCP	76
2.1	Apresentação detalhada da biblioteca “jamod”	76
2.2	Implementação de projeto de teste	77
2.3	Execução e análise do projeto final	79
3	Conclusões	83
C.	Anexo C - Especificação de Requisitos Alto Nível	84
1.	Introdução	86
1.1	Objetivos do documento	86
1.2	Âmbito	86
1.3	Contexto	86
1.4	Ambiente Técnico	86
1.5	Acrónimos e Definições	87
1.6	Visibilidade do Documento	88
2.	Necessidades	88
3.	Modelo do sistema	90
4.	Requisitos	92
5.	Matriz das Necessidades/Requisitos.....	96
6.	Diagrama de Casos de Uso.....	97
D.	Anexo D – <i>Mockups</i> para a interface Web do iMod	98
E.	Anexo E - Plano de Testes da interface Web do iMod.....	105
1.	Sessão de utilizador e navegação entre páginas	105
2.	Página de Monitorização.....	107
3.	Página de Configuração de Servidor	107
4.	Página de Configuração de Canais	118
5.	Página de Configurações Gerais	124
F.	Anexo F - Proposta de API do iHub.....	126
1	Introdução	128

2 Proposta	129
2.1 <i>Request</i> (pedido).....	129
2.2 <i>Response</i> (resposta).....	130
2.3 Estruturas (<i>keys</i>) JSON envolvidas nos pedidos e respostas.....	130
2.3.1 <i>Request</i>	130
2.3.2 <i>Response</i>	133
3 Conclusão	136
G. Anexo G - Arquitetura de interface Web do iMod.....	137
1. Interface de utilizador	137
2. Lógica	139
3. Persistência e Serviços	141
H. Anexo H - Manual de utilizador da interface Web do iMod	142
1. Introdução	144
2. Abordagem geral	144
3. Configuração de interface Web	145
4 Utilização	146
4.1 <i>Login</i>	146
4.2 <i>Mensagens de interface</i>	147
4.3 <i>Monitor</i>	148
4.4 <i>Server Configuration</i>	150
4.5 <i>Channels Configuration</i>	153
4.6 <i>General Configuration</i>	155
4.7 <i>Data Explorer/Reports/Alarms</i>	156
5 Considerações	158

ÍNDICE DE FIGURAS

Figura 1 - Logotipo da ISA.....	17
Figura 2 - Dispositivo iMod com antena GPRS.....	19
Figura 3 – Representação do Sistema de integração do iMod (DO: <i>Digital Output</i> – DI: <i>Digital Input</i>).	22
Figura 4 - Ciclo de desenvolvimento do Scrum	25
Figura 5 - Progama Jira, mostrando a gestão de tarefas divididas por <i>sprints</i> . Estas podem distinguir-se como "por fazer", "em progresso" ou "feita".	26
Figura 6 - <i>Burndown Chart</i>	27
Figura 7 – As várias camadas que representam as redes de comunicação, com exemplos protocolares e de <i>hardware</i> e materiais que se podem usar.	30
Figura 8 – Forma genérica como é feita a comunicação de uma central para um local remoto [fonte: referência 37].	30
Figura 9 – Exemplo de vista principal de sistema SCADA [fonte: ISA, interface SCADA de uma rede de água].....	31
Figura 10 – Exemplo de vista detalhada de sistema SCADA [fonte: ISA, interface SCADA de uma rede de água].....	32
Figura 11 – Uma rede de estações supervisionada e controlada por um sistema SCADA [fonte: ISA, interface SCADA de uma rede de água].	32
Figura 12 - NPE Search. Neste caso, foi encontrado um dispositivo iMod.	33
Figura 13 - NX Dynamics: página de <i>inputs/outputs</i> do iMod.....	34
Figura 14 - NX Dynamics: página de estado de sistema do iMod.....	34
Figura 15 - Acesso ao iMod via Telnet	35
Figura 16 – Ficheiro de configuração MainConfig.xml	37
Figura 17 – Interação via Modbus TCP com o iMod	38
Figura 18 - Modem GSM	40
Figura 19 - Arquitetura geral com foco no âmbito de desenvolvimento	50
Figura 20 - Conjunto de soluções a introduzir no iMod.....	51

Figura 21 - Diagrama representativo das várias componentes da camada de interface de utilizador e suas respectivas interações (ver legendas do diagrama).....	56
Figura 22 - Ambiente de teste de integração	61
Figura 23 – Transação <i>Master-Slave</i>	75
Figura 24 – Representação de uma simples ligação dos 2 módulos.....	75
Figura 25 – Aceder ao iMod na diretoria onde se encontra o <i>Slave</i>	79
Figura 26 – Execução do <i>Slave</i>	80
Figura 27 - Listagens de processos correntes	80
Figura 28 - Configuração da execução da aplicação Master	81
Figura 29 – Execução do Master	82
Figura 30 - <i>Mockup</i> para a página <i>Login</i>	98
Figura 31 - <i>Mockup</i> para a página <i>Monitor</i>	99
Figura 32 - <i>Mockup</i> para a página <i>Server Configuration</i>	100
Figura 33 - <i>Mockup</i> para a página <i>Channels Configuration</i> . As cruzes vermelhas significam que ainda estavam para ser definidos os atributos em questão para determinados tipos de parâmetros de canais.	101
Figura 34 - <i>Mockup</i> da página <i>Data Explorer</i>	102
Figura 35 - <i>Mockup</i> da página <i>Reports</i>	103
Figura 36 - <i>Mockup</i> da página <i>Alarms</i>	104
Figura 37 - Ficheiro de configuração de interface Web	145
Figura 38 – Esta é a página de <i>login</i> . É apresentada quando a sessão de utilizador não se encontra aberta.	146
Figura 39 – Para fazer login é necessário introduzir o nome de utilizador e a palavra-passe nos respetivos campos, clicando no botão “Login” para validar a operação.	147
Figura 40 – É mostrado o aviso “ <i>Warning: Username and/or password lefting!</i> ” quando um dos campos das credenciais está vazio após a submissão de <i>login</i> . Quando pelo menos uma das credenciais não está correta, é mostrado o aviso “ <i>Warning: Wrong username and/or password!</i> ”.	147

Figura 41 – Página <i>Monitor</i> com valores de dispositivos monitorizados e um botão para exportação de configurações e <i>logs</i>	149
Figura 42 – Neste caso, não foi possível obter valores por não se ter conseguido conectar com o protocolo iHub a fim de obter valores dos parâmetros mapeados.	149
Figura 43 – Página <i>Server Configuration</i> , é visível a configuração do tempo do servidor, um teste de conexão a servidor e a configuração geral do servidor.	150
Figura 44 – Ao clicar num dos botões de submissão para efetuar uma operação, é mostrado um painel informativo indicando para esperar.	151
Figura 45 – É mostrado o painel de sucesso de operação, que neste caso corresponde à configuração do tempo de servidor.....	151
Figura 46 – É mostrado o painel de aviso, que indica que não foi possível efetuar a configuração do tempo do servidor.	152
Figura 47 - Neste caso, não foi possível obter valores de configuração principais do protocolo iHub.	152
Figura 48 – Configuração de portas série e exportação de configuração	153
Figura 49 – Página <i>Channels Configuration</i> , onde possível ver uma tabela com os parâmetros a configurar, podendo-se escolher os seus tipos e editar vários atributos dos respetivos parâmetros.	154
Figura 50 – Contém um botão no final para exportação de configuração	154
Figura 51 – Página <i>General Configuration</i> , onde é possível a configuração do mapeamento de valores do equipamento.	155
Figura 52 – Sucesso na configuração do mapeamento. É mostrado um aviso quando existem valores de índices repetidos no mapeamento.....	155
Figura 53 – Página <i>Data Explorer</i> no estado atual.....	156
Figura 54 – Página <i>Reports</i> no estado atual.....	156
Figura 55 - Página <i>Alarms</i> no estado atual	157

ÍNDICE DE TABELAS

Tabela 1 - Matriz de Rastreabilidade de Necessidades/Requisitos	46
---	----

ABREVIATURAS

API – Application Programming Interface

ARM - Acorn RISC Machine

AVAC – Aquecimento, ventilação e ar condicionado

CSV - Comma-separated values

DHCP - Dynamic Host Configuration Protocol

GCC – GNU Compiler Collection

GNU - É um acrónimo recursivo de: GNU is Not Unix

GPL - Gás de Petróleo Liquefeito

GPRS - General Packet Radio Service

GSM - Global System for Mobile Communications (GSM: originalmente, Groupe Special Mobile)

GUI - Graphical User Interface

HTML – HyperText Markup Language

HTTP – Hypertext Transfer Protocol

IDE - Integrated Development Environment

IP – Internet Protocol

ISA – Intelligent Sensing Anywhere

ISEC – Instituto Superior de Engenharia de Coimbra

JSON – JavaScript Object Notation

M2M – Machine to Machine Communication

MAC - Media Access Control

NPE - universal industrial computer from the Techbase company

OLE - Object Linking and Embedding

OPC - OLE for Process Control

PC - Personal Computer

PHP – acrónimo recursivo para "PHP: Hypertext Preprocessor", originalmente "Personal Home Page"

PLC - Programmable Logic Controller

RAM –Random Access Memory

RISC - Reduced Instruction Set Computer

SCADA - Supervisory Control And Data Acquisition

SIM - Subscriber Identity Module

SMS - Short Message Service

SNMP – Simple Network Management Protocol

SVN - Subversion

TCP – Transmission Control Protocol

UI – User Interface

UML – Unified Modeling Language

URL - Uniform Resource Locator

USB - Universal Serial Bus

XAMPP - O nome provem da abreviação de X (para qualquer dos diferentes sistemas operativos), Apache, MySQL, PHP, Perl

XML – Extended Markup Language

1. INTRODUÇÃO

Neste capítulo contextualiza-se o estágio apresentando-se a empresa onde foi realizado, a descrição do projeto, os seus objectivos e os contributos para a sua realização.

1.1 ÂMBITO

A ISA é uma empresa de referência no mercado da telemetria, sendo o *core business* a telemetria de tanques de GPL industriais. Para além destas aplicações, a ISA disponibiliza um leque variado de soluções de telemetria que permitem a gestão inteligente de frotas, leitura automática de contadores e a eficiência energética.

No âmbito da monitorização e controlo de redes de abastecimento de água e iluminação pública a ISA tem o objetivo de integrar nas suas plataformas equipamentos de terceiros, tanto controladores como dispositivos periféricos.

No entanto, surgiu a oportunidade de aplicar esta tecnologia no âmbito da monitorização e controlo de equipamentos de uma instituição bancária, de modo a permitir a redução de consumo energético, e, conseqüentemente, os custos a ele associados.

1.2 APRESENTAÇÃO DA EMPRESA

A ISA - Intelligent Sensing Anywhere é uma empresa premiada e reconhecida internacionalmente, especializada na área da telemetria de comunicações M2M. É líder em diferentes segmentos desse mercado. Fundada em 1990 como *spin-off* da Universidade de Coimbra, a ISA conta com cerca de 100 colaboradores, escritórios em França, Espanha, Alemanha, Irlanda e Brasil, estando também presente em outros países através de agentes locais. O centro de I&D encontra-se em Portugal, onde cerca de metade da equipa da ISA se dedica permanentemente à inovação e desenvolvimento de produtos pioneiros.

As principais áreas de aplicação são:

- Telemetria Óleo & Gás;
- Soluções de Gestão de energia;
- Gestão remota de bens;
- Sistemas de Gestão integrada de Edifícios;

- Solução de Gestão remota para o ambiente;
- Soluções Médicas e de cuidados de saúde;

De entre todas estas áreas é de destacar a área de energia que inclui uma solução completa com sensores, contadores e toda a gama de dispositivos inovadores que estabelecem uma rede capaz de monitorizar à distância diversos parâmetros, tais como o consumo de água, gás e eletricidade, qualidade do ar, entre muitas outras funcionalidades.

A ISA tem várias soluções internas para a gestão de energia, sendo que é feita investigação de meios para tornar a recolha e processamento de dados de forma mais eficiente de modo a ir de encontro à expectativa dos clientes, integrando deste modo as soluções criadas usando tecnologias diversas. O controlador iMod é um exemplo de tecnologia externa que visa melhorar a qualidade de resposta.

O presente estágio está incluído na área das soluções de gestão de energia, chamada ISA Energy, servindo como controlo de eficiência energética para redução de consumos energéticos em edifícios. Além disso, o iMod utilizada no estágio para integração com soluções ISA.



Figura 1 - Logotipo da ISA

1.3 OBJETIVOS DO ESTÁGIO

O objetivo do estágio consistiu na integração do controlador iMod da Techbase nas plataformas ISA, devido ao facto deste equipamento ter várias funcionalidades essenciais para as soluções ISA (Anexo A – Proposta de Estágio).

Para a integração entre o iMod e as plataformas ISA é necessário conhecer o funcionamento do iMod e das plataformas ISA, de forma a escolher a estratégia que melhor se adapte à integração dos dois sistemas. O desenvolvimento de um módulo que monitorize o sistema é essencial e parte integrante deste do presente trabalho.

1.4 ORGANIZAÇÃO E TEMAS ABORDADOS NO PRESENTE RELATÓRIO

O presente documento encontra-se dividido em nove capítulos. Inclui também referências bibliográficas e um conjunto de anexos com informação complementar para ajudar a melhor compreender o problema e o trabalho realizado.

Após este capítulo de introdução (Capítulo 1), no Capítulo 2, Análise do Problema, apresenta-se o problema em detalhe, num contexto global e no contexto do estágio.

No Capítulo 3, Plano de Trabalhos do estágio, é introduzida a metodologia de desenvolvimento usada durante o projeto, bem como a sua implementação. Segue-se o planeamento inicial para o projeto bem como a sua justificação.

No Capítulo 4, Fase de Integração, é feita a descrição do trabalho inicial executado antes de o projeto propriamente dito se ter iniciado, por forma a integrar-me na metodologia, nas tecnologias de desenvolvimento usadas pela empresa e nas novas tecnologias de terceiros.

No Capítulo 5, Análise Tecnológica, pretende-se analisar o estado do projeto e as tecnologias a usar na implementação do sistema.

No Capítulo 6, Especificação, descrevem-se os vários tipos de requisitos da solução a implementar, justificando-se também as tecnologias adotadas.

No Capítulo 7, Arquitetura e Implementação da Plataforma, começa-se por apresentar a arquitetura global da aplicação. De seguida é efetuada uma descrição pormenorizada da solução desenvolvida, na qual se referem os vários aspetos e detalhes do modelo de dados e da interface.

No Capítulo 8, Verificação e Validação, são apresentados os testes e verificações a que o *software* foi sujeito de forma a garantir que estava dentro dos parâmetros de qualidade desejados.

No Capítulo 9, Avaliação de Resultados, faz-se uma análise e discussão sobre os resultados obtidos no estágio.

2. ANÁLISE DO PROBLEMA

Neste capítulo é feita a apresentação do problema num contexto global. São também apresentados alguns conceitos fundamentais para uma melhor compreensão do problema e do projeto.

2.1 ÂMBITO INICIAL

No início do estágio, o âmbito do projeto era, por assim dizer, ainda um pouco vago, no sentido em que existiam ainda muitas decisões e opções em jogo. Existia uma forte componente experimental associada a um plano adstrito ao setor de investigação da ISA.

2.1.1 O iMod

O iMod é um dispositivo que consiste num controlador industrial com funcionalidades para monitorização, telemetria de controlo remoto e registo de dados, que pode ser parametrizado com auxílio de protocolos (alguns personalizados). Existe opcionalmente a versão com antena GPRS integrada.



Figura 2 - Dispositivo iMod com antena GPRS

2.1.1.1 Porquê o iMod?

Existem várias razões pela qual a ISA optou pelo dispositivo iMod. Eis algumas das principais razões:

- É um dispositivo para uso de comunicação remota com baixo consumo de recursos e um processador de baixo consumo energético, bastante fiável durante longos períodos de tempo;

- Contém entradas físicas para *input/output* que são essenciais, tais como portas série, uma porta Ethernet e opcionalmente um porta para utilização de antena GPRS;
- Contém comandos e bibliotecas de modo a poderem desenvolver-se soluções de *software* destinadas a execução no iMod, em diversas linguagens (ex: Java, Python, PHP, Bash);
- Tem uma boa relação qualidade/preço;
- Permite obtenção e controlo de dados em tempo real, isto é, assincronamente.

2.1.1.2 Pré-estudo disponibilizado

Antes de o estágio ter tido início, a ISA tinha adquirido anteriormente alguns dispositivos iMod a fim de realizar uma exploração preliminar deste dispositivo. O responsável por estes testes elaborou um documento que indica como instalar e aceder ao conteúdo de um dispositivo iMod, seja pela interface Web disponibilizada (ver adiante NX Dynamics) ou via FTP.

O referido documento foi disponibilizado, tendo sido bastante útil para a realização dos primeiros contactos com o dispositivo iMod.

2.2 INVESTIGAÇÃO DO IMOD E TECNOLOGIAS

Após o pré-estudo, foi realizado o estudo ao iMod, em que este foi explorado, assim como as suas ferramentas.

No sentido de entender em que consistem soluções de monitorização e controlo, como parte importante das soluções ISA, foi abordado esse tipo de soluções e tecnologias associadas: estudo de soluções de monitorização e controlo ISA; sistemas SCADA; protocolo Modbus; comunicação por porta série (GPRS e GSM).

2.3 DESENVOLVIMENTO INTEGRADO EM PROJETO PARA CLIENTE

Decorrido este período inicial, o projeto de controlo e comunicação com o iMod foi agregado a um projeto destinado a um cliente externo, no caso, uma instituição bancária.

Tendo em conta que a ISA já possuía experiência neste tipo de soluções, o desafio era a integração do iMod nas soluções já existentes, visando a melhoria e eficiência das soluções ISA para satisfazer o cliente.

O objetivo consistia na implementação de uma solução ISA, aproveitando um *firmware* protocolar já existente, a ser remodelado (protocolo iHub), e com vista a integração com o iMod.

2.3.1 Visão e âmbito do projeto

Fornecimento e configuração de um sistema centralizado de recolha de dados e dispositivos remotos para recolha e submissão de dados no âmbito da supervisão do consumo de energia na rede bancária, de modo a permitir maior eficiência energética.

Isto é, o sistema deve permitir a monitorização do consumo de energia do banco de forma a ser reduzido, evitando desperdícios, promovendo uma maior poupança energética e, conseqüentemente, sendo económica para a empresa.

O objetivo por parte do banco consiste em reduzir consumos elétricos através de um controlo adequado de todos os sistemas elétricos utilizados nas agências.

2.3.2 Principais entidades do sistema

Ao analisar as necessidades no contexto deste projeto, criou-se um diagrama para representação geral da arquitetura do sistema (Figura 3). Na figura, para simplificação, é representada apenas uma agência bancária, embora consista em toda uma rede de agências bancárias.

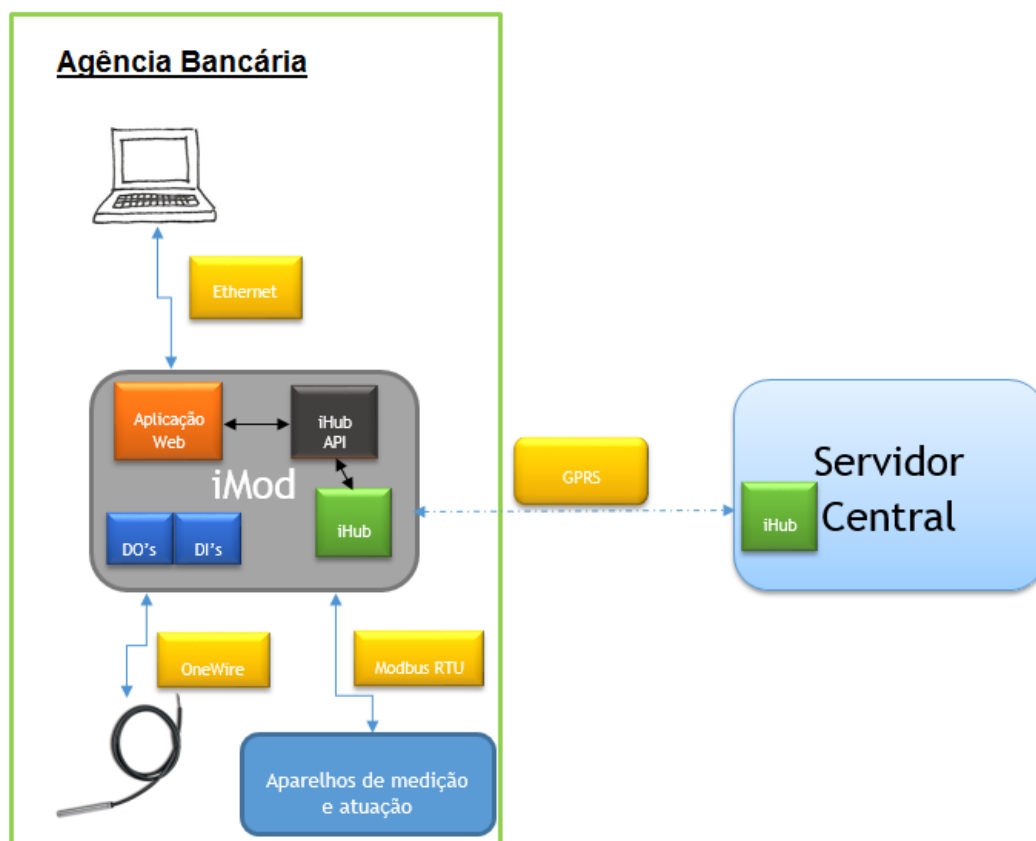


Figura 3 – Representação do Sistema de integração do iMod (DO: *Digital Output* – DI: *Digital Input*).

As entidades e as suas relações são as seguintes:

- O iMod inicia a comunicação com o servidor central usando o protocolo iHub via GPRS, pelo que o iMod tem de ser o de versão com antenas de GPRS. A instituição bancária optou pela utilização de GPRS ao invés da rede Ethernet das agências, pois não pretende incluir uma entidade externa na rede das agências por motivos de segurança, embora a solução assim fique mais dispendiosa devido aos custos com as comunicações GPRS;
- Por conseguinte, é aberto um canal de comunicação para transmissão de dados através da conexão acima referida;
- O iMod lê e escreve parâmetros dos vários aparelhos de medição (sensores, contadores, entre outros), através do protocolo Modbus RTU via porta série;

- A comunicação entre o sensor de temperatura (utilizado para medir a temperatura ambiente de uma agência bancária) e o iMod é conseguida através do protocolo OneWire;
- Pode aceder-se a uma interface Web do iMod através de um cliente Web (*browser*) de um computador. Essa interface permite visualizar e controlar as configurações de um protocolo de monitorização e controlo (iHub). Para tal, será desenvolvida uma API de modo a criar uma ponte de ligação entre a interface Web e o protocolo (será apresentado maior detalhe sobre a API mais adiante);
- O iMod e computador ficam ligados por cabo Ethernet a fim de estabelecer a comunicação necessária para o acesso à interface Web, a fim do instalador da ISA poder analisar os valores recolhidos, configurações e *logs*, para que eventualmente possa alterar configurações. Basicamente, a interface é necessária no âmbito de manutenção de cada iMod das várias agências bancárias.

2.3.3 Descrição Geral do Projeto de Estágio

O projeto de estágio tem como objetivo o desenvolvimento de uma aplicação Web, que apresenta ao utilizador todas as informações relevantes obtidas de um protocolo desenvolvido pela ISA (iHub), permitindo a sua configuração.

A obtenção e configuração da informação apresentada ao utilizador são feitas através de uma API do protocolo, que deve ser igualmente desenvolvida. Essa API é uma interface, que serve para a aplicação Web se tornar mais independente do protocolo de recolha de dados (iHub) e ser mais intuitiva na obtenção e configuração de dados.

3. PLANO DE TRABALHOS DO ESTÁGIO

Este capítulo apresenta as metodologias e ferramentas de trabalho e desenvolvimento do estágio, do ponto de vista da gestão e organização do mesmo.

3.1 METODOLOGIA DE DESENVOLVIMENTO

A metodologia de desenvolvimento utilizada no âmbito deste projeto é a metodologia Scrum.

3.1.1 Descrição do Scrum

O Scrum é uma metodologia ágil para gestão e planeamento de projetos. As metodologias ágeis de desenvolvimento de *software* são iterativas, ou seja, o trabalho é dividido em iterações, que são designadas por *sprints*, no caso do Scrum.

Esta metodologia caracteriza-se pela elevada produtividade das equipas envolvidas, obtida através de diversas técnicas que vão desde a construção de um sólido espírito de equipa, a uma forte autorresponsabilização da mesma, e uma consistente componente de melhoria contínua dos processos de desenvolvimento.

As funcionalidades a serem implementadas durante o projeto são mantidas numa lista que é conhecida como *product backlog*. No início de cada *sprint*, faz-se um *sprint planning meeting*, ou seja, uma reunião de planeamento na qual o *product owner* prioriza os itens do *product backlog* e a equipa seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia. As tarefas alocadas a um *sprint* são transferidas do *product backlog* para o *sprint backlog*.

A cada dia de um *sprint*, a equipa faz uma breve reunião (normalmente de manhã), chamada *daily scrum*. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

No final de um *sprint*, a equipa apresenta as funcionalidades implementadas num *sprint review meeting*. Finalmente, faz-se um *sprint retrospective* e a equipa parte para o planeamento do próximo *sprint*. Assim reinicia-se o ciclo.

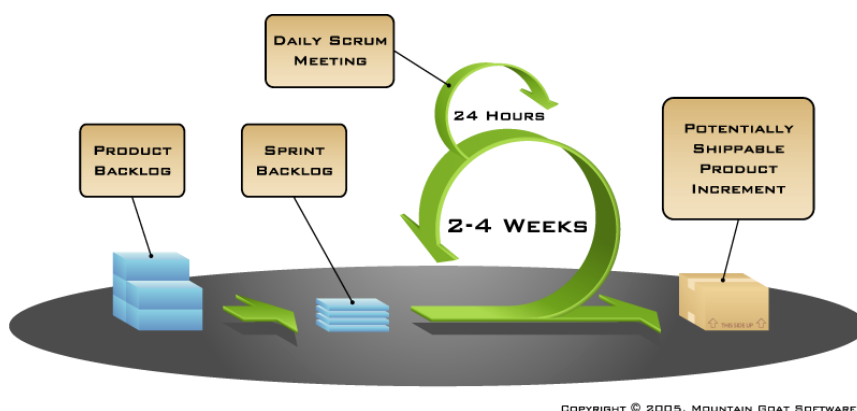


Figura 4 - Ciclo de desenvolvimento do Scrum

3.2 APLICAÇÃO DO SCRUM NO PROJETO DE ESTÁGIO

Neste ponto será explicada efetivamente a aplicação desta metodologia ao projeto.

3.2.1 *Pre-Game* (Análise)

A fase de *Pre-Game* corresponde à *sprint 0* do Jira (Ferramenta de Gestão de Projeto - Figura 5 e Figura 6), onde o estagiário se foi familiarizando com as ferramentas e integrando no projeto.

3.2.1.1 *Product BackLog* (*Product owner*)

A responsável pelo *Product Backlog*, ou seja, o *Product Owner*, era a coordenadora Eng. Rita Carreira, que transmitia as necessidades e requisitos essenciais para o decorrer do projeto.

3.2.1.2 Ferramentas de Suporte

Ao longo do estágio foram usadas várias ferramentas, mas devem destacar-se duas delas, visto terem sido de vital importância para o sucesso do projeto e suporte do Scrum: a ferramenta JIRA e o SVN.

- **SVN** – Foi usado um repositório com todos os artefactos gerados durante o projeto: este foi um repositório SVN. Um repositório SVN permite ter um histórico de alterações de código e de documentação;
- **JIRA** – Ferramenta de gestão de projeto, onde é possível ver todas as tarefas atribuídas em cada um dos *sprints*, atrasos, etc. Esta ferramenta foi usada diariamente, as tarefas foram sendo atualizadas com os *logs* dos tempos efetuados (Figura 5 e Figura 6);

- **Balsamiq Mockups** – Para construir protótipos de interface com o utilizador de modo a validar os requisitos do sistema;
- **Enterprise Architect** - Uma ferramenta que oferece capacidade de gestão para modelizar soluções de *software* nos seus vários domínios, desde a análise de requisitos ao desenho detalhado. Foi utilizada para construir um diagrama de Casos de Uso;
- **Visio** – Uma ferramenta que permite criar todo o tipo de diagramas de UML, esquemas de redes, entre outros.

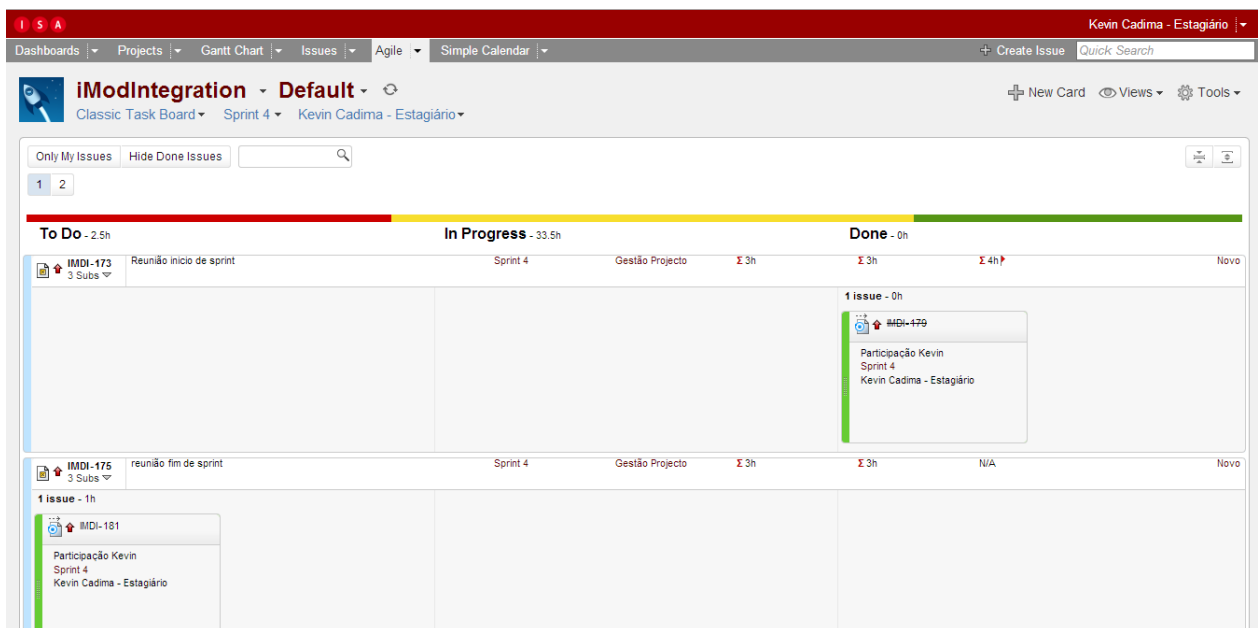


Figura 5 - Progama Jira, mostrando a gestão de tarefas divididas por *sprints*. Estas podem distinguir-se como "por fazer", "em progresso" ou "feita".

3.2.1.3 Equipa e Papéis

A equipa do projeto de integração com o iMod consiste no autor do presente documento, um estagiário de mestrado de engenharia eletromecânica e a coordenadora de estágio, que é *Product Owner* e *Scrum Master* do projeto em questão.

3.2.1.4 Datas

A duração do projeto coincidiu com o tempo de estágio: de 8 de outubro de 2013 a 27 de junho de 2014.

3.2.2 Game (Análise, Desenho, Construção, Pré-produção)

A *Game* iniciou-se com uma parte experimental, a que correspondem as *sprints* 1 e 2. As partes desde a análise à aceitação enquadram-se nas *sprints* 3 a 5, que grosso modo correspondem a: *sprint* 3 - análise de requisitos e arquitetura; *sprint* 4 – implementação; *sprint* 5 - verificação, validação e aceitação.

3.2.2.1 Reunião de Planeamento *Sprint*

No início de cada *sprint* havia uma reunião em que analisávamos a *sprint* que tinha terminado, as tarefas marcadas no Jira como “feitas”, “a decorrer” e “por fazer”, de modo a dar seguimento à *sprint* atual.

3.2.2.2 *Sprint*

O projeto foi dividido em 6 *sprints* que representavam fases de marco no projeto, representadas por decisões que foram surgindo ao longo do tempo.

A figura 8 mostra os *sprints* existentes e permite a visualização do *burndown chart* de um *sprint*, que consiste numa representação gráfica das estimativas, tempos restantes e tempos consumidos das tarefas atribuída no início da respetiva *sprint*.

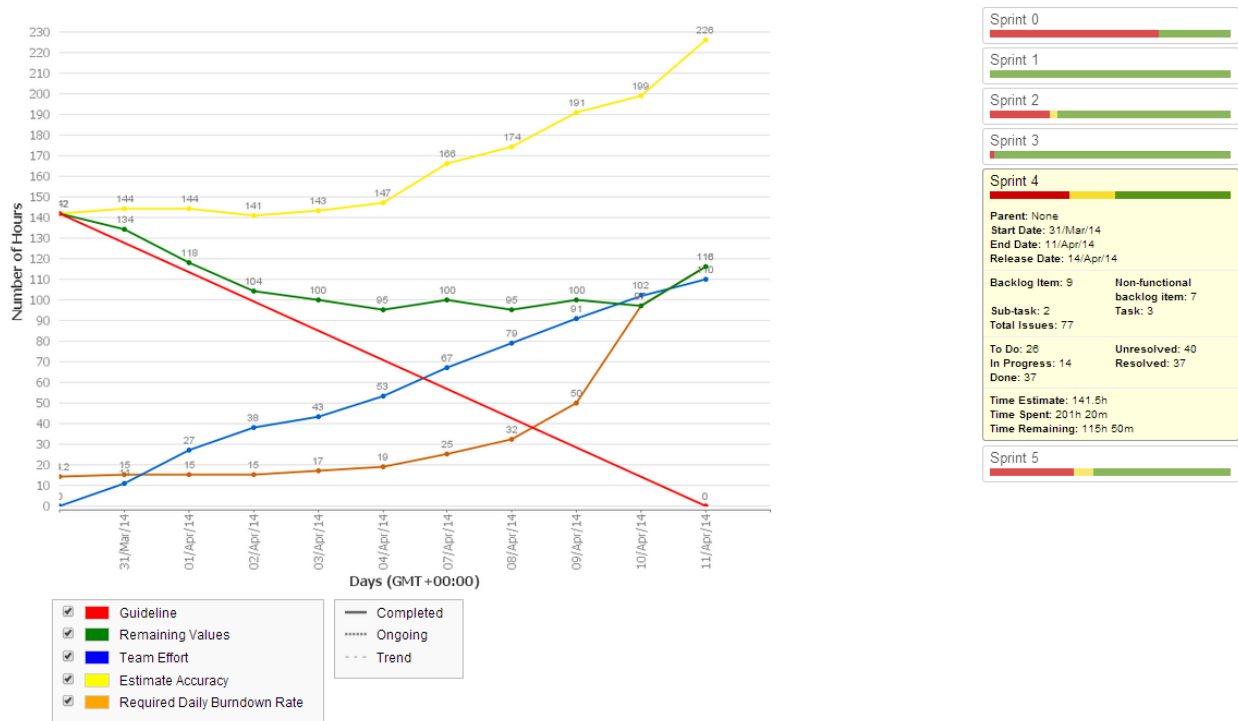


Figura 6 - Burndown Chart

3.2.2.3 Reunião diária

Todas as manhãs, havia reuniões curtas (*daily scrum*) para fazer um ponto de situação, tirar dúvidas, relatar as tarefas realizadas no dia anterior e o plano de trabalhos para o presente dia.

3.2.2.4 Revisão

No final de cada *sprint* havia uma reunião em que era feita uma revisão das tarefas no Jira para a nova *sprint*, como por exemplo, ver se alguma tarefa que deveria ter sido fechada está efetivamente fechada.

3.2.2.5 Retrospectiva

Feita em conjunto com a revisão, consistia na apresentação dos artefatos produzidos no decorrer do *sprint* em questão à coordenadora (*Product Owner*), como por exemplo, rever o documento de requisitos ou utilizar uma aplicação no estado atual.

3.2.3 *Post-Game* (Operacionalização)

Não existe *Post-Game*, porque apesar de ter sido dado um término ao trabalho realizado no projeto do estágio, que inclui muitos artefactos válidos, tais artefactos não estão ainda aptos a instalar no cliente, pois a sua conclusão envolve outra equipa de desenvolvimento no que toca à parte protocolar, pelo que o produto não está em versão final.

4. FASE DE INTEGRAÇÃO E INVESTIGAÇÃO

Na fase inicial, até surgir uma visão e âmbito mais concretos, decorreu a fase de contextualização com o iMod e a aprendizagem de tecnologias associadas a este tipo de projetos, e que se encontram em algumas soluções da ISA.

4.1 CONTEXTUALIZAÇÃO

No início, foi lida alguma documentação genérica sobre soluções e projetos colocados em campo pela ISA, para familiarização com as tecnologias da empresa e de certa forma com o âmbito do projeto.

Foram apresentados os processos de controlo de qualidade da empresa e as plataformas para aceder a esses processos e obter os *templates* para a documentação necessária.

Foi lida documentação variada sobre o iMod (dispositivo pertencente à empresa polaca Techbase).

Foi estudado o protocolo Modbus, que é bastante usado para a leitura e escrita de dados em ambiente industrial.

Estudou-se o conceito de PLC – um tipo de dispositivo que utiliza o protocolo Modbus – e que funciona como um ponto direto para recolha e manipulação de valores digitais, isto é: um dispositivo iMod pode estar ligado a vários PLC's a fim de recolher e manipular uma grande quantidade de dados.

Os equipamentos eletrónicos associados aos PLC podem corresponder a bombas de água numa estação elevatória; sensores de temperatura; sensores de pressão; válvulas; iluminação; ar condicionado; ou seja, uma infinidade de equipamentos essenciais para vários setores profissionais (extração, produção, indústria, distribuição, etc.).

Foi adquirida informação sobre vários termos relacionados com redes de comunicação, tais como FTP, SNMP, IMEI e NPE (em que se enquadra o iMod).

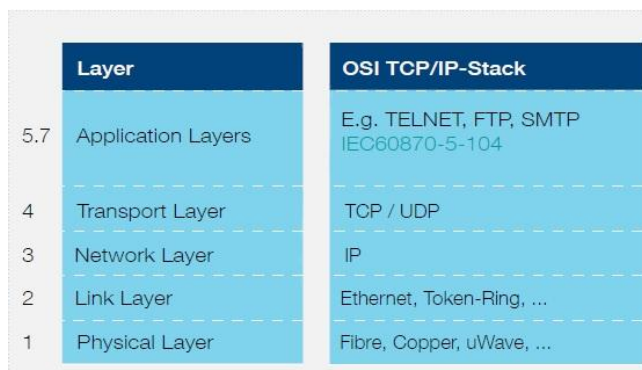


Figura 7 – As várias camadas que representam as redes de comunicação, com exemplos protocolares e de *hardware* e materiais que se podem usar.

Na Figura 8 [37], pode-se ver de forma genérica como é feita a comunicação de uma central com um local remoto ou, especificamente, em que um PC ou servidor, que é mestre na ligação, comunica com um dispositivo de controlo para recolher e/ou induzir dados (poderia ser o iMod), que por sua vez está ligado direta ou indiretamente a PLC's, sensores, medidores, etc. para serem controlados e/ou deles serem obtidos valores.

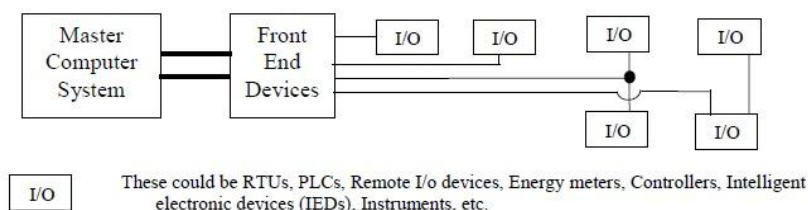


Figura 8 – Forma genérica como é feita a comunicação de uma central para um local remoto [fonte: referência 37].

Muitas configurações e dados contidos nos NPE - o iMod é um modelo de NPE - são em geral ficheiros de formato XML ou CSV.

Foi analisado um protótipo de *scripts* em Bash a serem experimentados no iMod com algumas das soluções da ISA, o que permitiu compreender a forma como os dados são gravados enquanto não tiverem sido enviados para um repositório remoto, e como os ficheiros são identificados de forma a não criar conflitos ou incoerências. Uma vez que se estava a lidar com *scripts* Bash, decidiu-se fazer um *script* Bash experimental a fim de permitir familiarização com este tipo de *scripts*.

4.2 SCADA

Teve-se em conta a possibilidade de ser necessário desenvolver uma interface SCADA para o corrente projeto de estágio, e, por isso, estudou-se esta tecnologia. A tecnologia SCADA é bastante utilizada nas empresas para monitorização e controlo remoto de indústrias, estações de distribuição, etc., e, evidentemente, em soluções da ISA.

SCADA é um tipo de sistemas bastante utilizado, pelo que existe um leque de ferramentas disponíveis de modo a implementar intuitivamente e de forma relativamente acessível uma solução SCADA à medida de cada empresa.

Para se poder vir a lidar e implementar algo em SCADA, instalou-se uma ferramenta chamada Movicon, já utilizada pela ISA.

A aplicação Movicon é uma ferramenta bastante utilizada atualmente para criar sistemas SCADA, e nesse sentido foram visualizados tutoriais e criaram-se interfaces, permitindo ver a grande variedade de protocolos de comunicação e de dados (do mais alto nível ao mais baixo nível, do mais informático ao mais industrial); o potencial gráfico e a forma intuitiva como se podem criar interfaces para visualização e controlo de equipamentos.

Na Figura 9 pode ver-se um exemplo de sistema SCADA, que representa a vista principal que permite aceder à localização de várias estações. É intuitivo porque é representado por um mapa, em que as cores laranja e vermelho representam diferentes níveis de alarmes, que podem ser encontrados nas regiões em questão.



Figura 9 – Exemplo de vista principal de sistema SCADA [fonte: ISA, interface SCADA de uma rede de água].

Na Figura 10 pode ver-se um exemplo de interface de sistema SCADA, que representa uma estação de depósito de água com uma bomba de elevação.

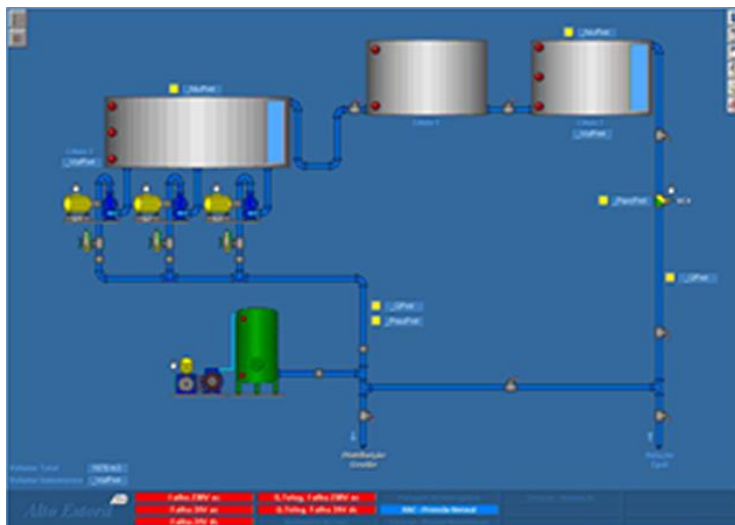


Figura 10 – Exemplo de vista detalhada de sistema SCADA [fonte: ISA, interface SCADA de uma rede de água].

Na Figura 11 pode ver-se uma rede de estações supervisionada e controlada por um sistema SCADA, que inclui um servidor conectado a vários PLC's para permitir a conexão a todo um conjunto de estações. Tendo em conta a abordagem do projeto de integração do iMod, um possível dispositivo iMod situar-se-ia, numa das estações (um para cada estação) que, por sua via, estaria ligado a um ou vários PLC's (por exemplo) a fim de poder controlar e recolher dados de vários equipamentos.

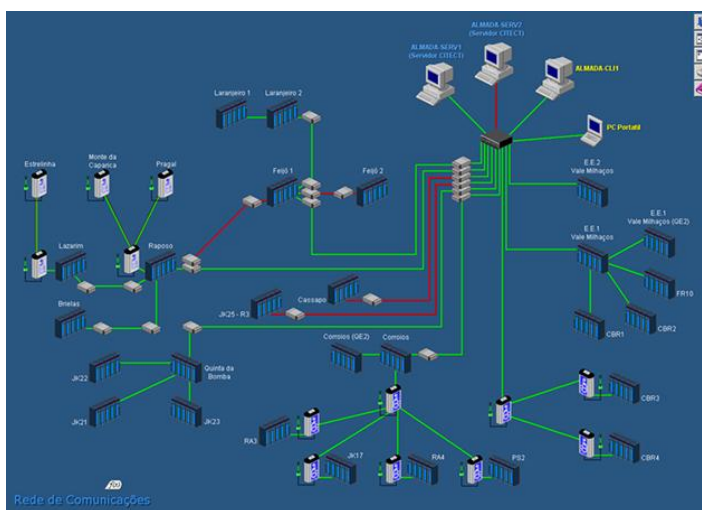


Figura 11 – Uma rede de estações supervisionada e controlada por um sistema SCADA [fonte: ISA, interface SCADA de uma rede de água].

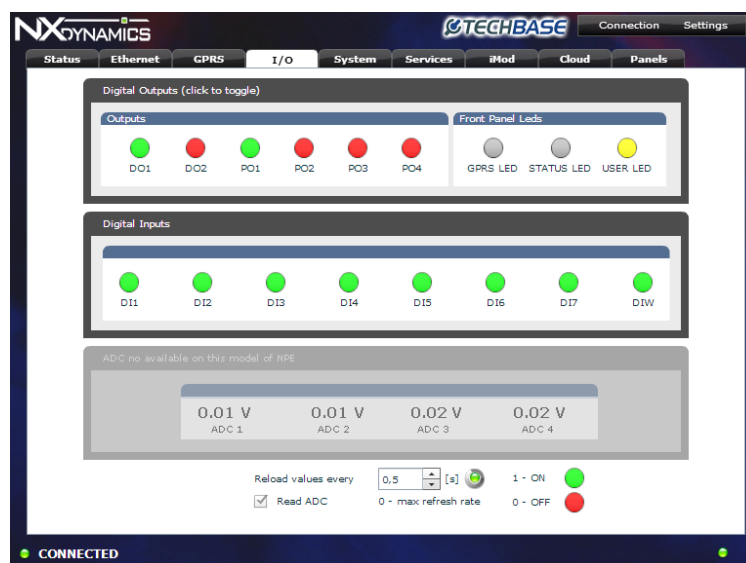


Figura 13 - NX Dynamics: página de *inputs/outputs* do iMod

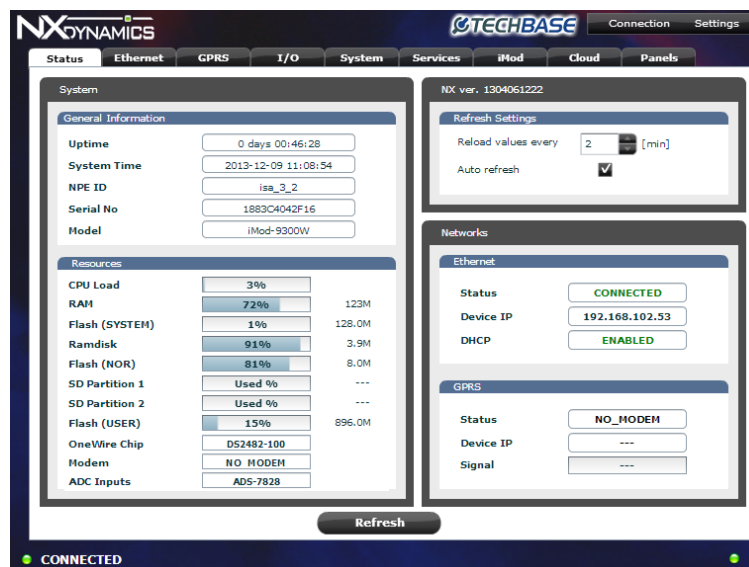
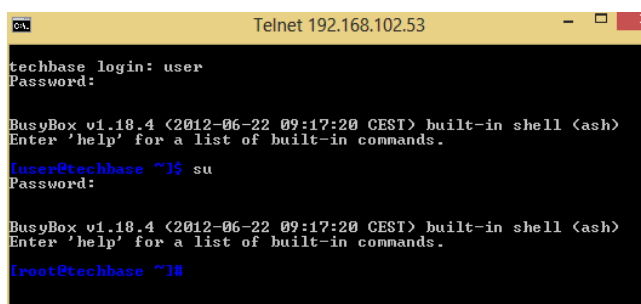


Figura 14 - NX Dynamics: página de estado de sistema do iMod

Como o iMod inclui o Linux Kernel como sistema operativo, decidiu-se instalar uma máquina virtual com o sistema operativo Linux Ubuntu no programa Virtual Box, a fim de se poder ter sempre à mão um ambiente experimental e acesso a documentação do Linux. Além do Linux Kernel, o iMod tem instalado um pack de comandos denominado MagicBox, que contém comandos básicos do Linux, assim como comandos proprietários da Techbase.

Aprendeu-se a aceder ao iMod via Telnet.



```
ca. Telnet 192.168.102.53
techbase login: user
Password:
BusyBox v1.18.4 (2012-06-22 09:17:20 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
[user@techbase ~]# su
Password:
BusyBox v1.18.4 (2012-06-22 09:17:20 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
[root@techbase ~]#
```

Figura 15 - Acesso ao iMod via Telnet

Para poder descobrir e manipular o conteúdo do iMod de forma segura, decidiu-se fazer um *backup* de todo o conteúdo (pastas e ficheiro) de 2 iMod's (disponíveis na altura). Já se conseguia aceder, abrir e colocar ficheiros individuais no iMod via FTP com o programa Notepad++, mas infelizmente não era o suficiente para fazer o *backup*, pois este processo não permite copiar recursivamente toda uma pasta ou conjunto de pastas.

Por isso decidiu investigar-se ferramentas que permitissem fazê-lo, o que levou a utilizar-se o WISE-FTP. Mais tarde, foi utilizada uma ferramenta gratuita equivalente, de bom desempenho, chamada WinSCP.

Uma vez que o iMod permite a execução de programas desenvolvidos em Java, optou-se por experimentar essa possibilidade instalando a IDE Eclipse proposta pela Techbase.

É possível desenvolver-se um programa, colocar as bibliotecas necessárias e exportar de modo a criar o executável no formato “.jar”. Assim é possível colocar um projeto Java compilado no iMod (numa pasta à escolha, utilizando acesso por FTP, bastando o Notepad++), podendo-se executá-lo apenas utilizando um comando para esse efeito (comando do JRE inerente ao iMod).

Podia-se parar o processo executado apenas procurando o ID do mesmo através de um comando e depois utilizando outro comando para terminar forçadamente o processo em questão, especificando-se o respetivo ID.

Para os testes foi desenvolvido um simples servidor TCP/IP, que depois evoluiu de forma a integrar o protocolo Modbus.

4.4 BIBLIOTECA MODBUS

Para entender melhor o protocolo Modbus e desenvolver uma solução experimental para correr no iMod, foi decidido que se procurasse uma biblioteca Java que implementasse o protocolo Modbus e respetivas transações de comunicação (Modbus-TCP).

Após se terem investigado bibliotecas existentes e se terem descartado algumas delas (por serem incompletas, licenciadas, sem documentação, para outra linguagem, ou que não correspondiam no geral à expectativa), encontrou-se uma adequada chamada “jamod” (para Java Modbus).

A biblioteca “jamod” é uma biblioteca gratuita, bem documentada, compatível e desenvolvida para soluções em Java, que contém a utilização do protocolo Modbus de forma intuitiva para vários tipos de comunicação (Mestre e Escravo por porta série, Mestre e Escravo por UDP, e Mestre e Escravo por TCP).

4.4.1 Modbus

O protocolo Modbus é dos protocolos mais antigos e mais utilizados em sistemas de automação industrial, em que é muito frequente a sua utilização para a comunicação entre PLC's e outros dispositivos.

4.4.2 Desenvolvimento experimental – cliente Modbus-TCP (Mestre) – servidor TCP

Experimentou-se criar um simples cliente Modbus-TCP (Mestre), isto é, um cliente TCP, destinado a enviar uma mensagem de protocolo Modbus e com a iniciativa de comunicar com o servidor TCP.

O servidor TCP utilizado nesta fase foi um simples servidor TCP corrido com o programa Hercules (nesta fase configurado em *localhost*). Correu-se o cliente a partir da IDE Eclipse.

A experiência correu com sucesso, pelo que foi possível verem-se os dados do pedido Modbus na consola do Hercules.

4.4.3 Criação de documento

No intuito de documentar a biblioteca “jamod” para futura utilização, foi criado um documento em que constam o âmbito, as razões que levaram a procurar e escolher esta biblioteca, assim como a descrição e apresentação técnica da mesma, mostrando um exemplo de projeto que utiliza a biblioteca (Anexo B - Procura de Biblioteca Modbus-TCP).

4.4.4 Desenvolvimento experimental – cliente Modbus-TCP (Mestre) – servidor Modbus-TCP (Escravo)

Após se ter tido o cliente Modbus-TCP (Mestre) mais completo no tipo de dados a ler e/ou escrever, implementou-se um servidor Modbus-TCP (Escravo), que estaria à espera de pedidos Modbus de clientes.

Esse servidor destinar-se-ia a ser colocado no iMod para ler/escrever num dos PLC's a ele ligados, utilizando-se uma configuração num XML para o efeito (MainConfig.xml – ver Figura 16), indicando os parâmetros e configurações associados a portas séries.

Por este motivo, teve-se de fazer uma função que fizesse *parsing* desse ficheiro, a fim de se poder obter os parâmetros a definir no conjunto de configurações do servidor Modbus-TCP. O tipo de parâmetro seria descrito no ID da *tag* no XML, para este poder ser intuitivamente definido no servidor. Conseguiu-se definir no servidor os parâmetros definidos no XML com sucesso.

```
<group name="parameters">

  <parameter type="int16">
    <id>100</id>
    <description>"Register"</description>
    <init-value>"0"</init-value>
    <source-channel channel-name="NPE_io" parameter-id="DO1"/>
    <access-channel channel-name="SNMP_Slave_1" parameter-id="100" />
  </parameter>

  <parameter type="int16">
    <id>101</id>
    <description>"Input Register"</description>
    <init-value>"985"</init-value>
    <access-channel channel-name="SNMP_Slave_1" parameter-id="101"/>
  </parameter>

  <parameter type="int16">
    <id>102</id>
    <description>"Register"</description>
    <init-value>"4256"</init-value>
    <access-channel channel-name="SNMP_Slave_1" parameter-id="102"/>
  </parameter>

  <parameter>
    <id>103</id>
    <description>"Digital Out"</description>
    <init-value>"1"</init-value>
    <access-channel channel-name="SNMP_Slave_1" parameter-id="103"/>
  </parameter>

  <parameter type="int16">
    <id>104</id>
    <description>"Register"</description>
```

Figura 16 – Ficheiro de configuração MainConfig.xml

Introduziu-se um ciclo que tentava fazer o *parsing* N vezes (parâmetro configurável) para cada parâmetro sempre que houvesse uma falha, no sentido de garantir que o servidor tinha carregado os parâmetros da configuração com sucesso.

O iMod tem disponível o comando “modmas”, que é um comando que implementa Modbus tendo o âmbito da configuração dos parâmetros no XML do iMod, ou seja, para ler e escrever valores de parâmetros, associados por exemplo a um PLC.

Como a biblioteca “jamod” (Anexo B - Procura de Biblioteca Modbus-TCP) implementa o Modbus de uma forma simples e fora do contexto do iMod, então teve-se de adaptar as classes que representam a leitura e escrita para os vários tipos de parâmetros, em que neste, se pôde utilizar o comando “modmas” no código seguindo cada caso. Basicamente, à funcionalidade Modbus TCP da biblioteca, complementaram-se as operações de atuação Modbus inerentes ao iMod para leitura e escrita de valores, a fim de atuar nos PLC’s, etc..

Experimentou-se primeiro com leitura, pois era mais fácil de implementar e de testar, e só então se fez o equivalente para escrita. Como a escrita de valor não era efetuada com sucesso sempre após a primeira operação, optou-se por fazer N vezes a escrita, para garantir que um valor tinha sido alterado. A falta de persistência na escrita de valores, pode dever-se a algum *delay* na operação (por exemplo, por causa de outra escrita ainda a decorrer), ou também quanto à forma como podem ser interpretados os dados: por exemplo, se for escrito o valor 37, a 1ª tentativa interpreta o valor como sendo 1 (de boleano verdadeiro), mas a 2ª tentativa já considera o valor 37.

Testou-se várias vezes, com vários tipos de parâmetros, para leitura e escrita. Ao fim de várias correções, conseguiu-se ter resultados satisfatórios.

Os resultados foram um sucesso, pois foi possível interagir com os dispositivos, tanto em leitura como em escrita, através dos respetivos parâmetros. Por exemplo, se um parâmetro boleano está com valor a 1 (verdadeiro - portanto deixando a respetiva luz do PLC acesa), é possível escrever o valor 0 (falso - atuando no PLC e apagando a referida luz).

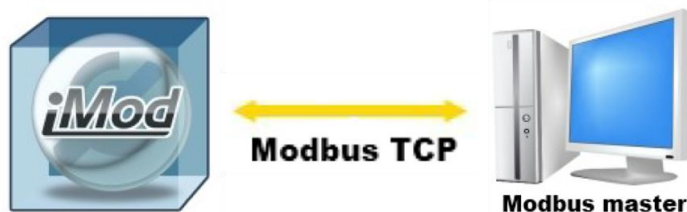


Figura 17 – Interação via Modbus TCP com o iMod

4.4.5 Desenvolvimento experimental – cliente Modbus-TCP (Escravo) – servidor Modbus-TCP (Mestre)

Entretanto, decidiu-se implementar uma versão “invertida” do protocolo Modbus-TCP, funcional apenas via Ethernet. Por “invertida”, quer-se significar que o servidor passa a ter IP dinâmico e contém uma lista de endereços IP dos vários clientes, que têm IP estático. O servidor é quem tem, por isso, a “iniciativa” de criar a comunicação.

Após a comunicação estabelecida, o cliente volta a ter o papel de mestre, uma vez que a iniciativa é-lhe devolvida, podendo fazer os vários pedidos de leitura/escrita de parâmetros através do protocolo Modbus-TCP.

4.5 COMUNICAÇÃO POR PORTA SÉRIE

Muita da comunicação é feita através de portas série, seja diretamente entre dois dispositivos, ou entre dois dispositivos remotos, através de um *modem* (de GPRS, de GSM, etc.).

Para entender como funciona, fizeram-se algumas experiências.

4.5.1 GPRS e pesquisa de biblioteca Java que implemente comunicação via GPRS

Com intenção de entender melhor o que é GPRS e como é que esse tipo de comunicação é implementado, estudou-se o assunto e procurou-se uma biblioteca e/ou código Java que implementasse esse tipo de comunicação.

Tentou-se aplicar esse tipo de comunicação no projeto Modbus-TCP invertido.

Basicamente, é aberto um *socket* TCP/IP que escreve para um canal de porta série anteriormente aberto para o *modem* GPRS.

4.5.2 Comandos AT

A comunicação por portas série, no que toca aos *modems* GPRS e GSM, utiliza comandos AT.

Por exemplo, enviando simplesmente o comando “AT”, é estabelecida comunicação com o *modem*, sendo retornada uma mensagem de sucesso ou insucesso da operação.

No caso acima referido, em relação ao GPRS, bastava abrir a comunicação para o *modem* GPRS com o comando “AT” e uma vez o canal aberto, bastaria colocar o *socket* TCP/IP a escrever para esse canal para ter uma ligação TCP/IP via GRPS (como alternativa ao Ethernet que é uma ligação física).

4.5.3 GSM

Fez-se um projeto Java no qual se pôde enviar um SMS para um número de telemóvel previamente definido aquando da sua execução, pelo que correu com sucesso. O envio de SMS's é bastante utilizado em sistemas de monitorização e controlo, a fim de responsáveis pela monitorização serem notificados quando algum evento ocorre.



Figura 18 - Modem GSM

Ligou-se um *modem* GSM, que contém um cartão SIM no seu interior, com o cabo série a um adaptador USB ligado ao PC.

Foi visto no Windows qual a porta série a ser utilizada, porque apesar de estar ligado por USB, este é apenas uma interface para a porta série. Configurou-se o ID da porta no código.

Já se tinha experimentado anteriormente com um cliente Serial no programa Hercules a entrar em contacto com o *modem* GSM, escolhendo a porta série em questão e enviando o comando AT “AT” e devolvendo mensagem de sucesso.

No código foi colocado, além do comando AT para fazer a ligação, também o comando AT para definir o número de telemóvel, o conteúdo do SMS e finalmente poder enviar o SMS.

Ao executar o programa com um número de telemóvel definido, pode-se efetivamente receber o SMS esperado.

4.6 Programas na linguagem C no iMod

Para permitir a utilização de *firmware* do iMod através de um protocolo (iHub) de que a ISA é proprietária e que foi desenvolvido em C, surgiu a necessidade de se poder programar em C aplicações para executar no iMod, investigando-se essa possibilidade.

O iMod é projetado para programação em Java, pelo que lhe foram removidas as bibliotecas C que são utilizadas normalmente no Linux. No entanto, elas encontram-se no seu predecessor e módulo base, chamado “NPE”, pelo que surgiu a ideia de embutir essas bibliotecas C no iMod. Mas estar-se-ia a colocar em risco a garantia e suporte da Techbase e, pior ainda, todo um sistema de revisão de instalação de pacotes no sistema operativo do iMod, que pode remover pacotes estranhos e colocar pacotes em falta ou com *links* corrompidos.

Como alternativa, pensou-se em compilar um programa em C com o GCC (o compilador de C no Linux) feita no Linux da máquina virtual, e depois corrê-lo no iMod. Foi criado um simples programa em C, foi compilado como pretendido, e tentou-se executá-lo no iMod, mas não foi possível, não se sabendo concretamente qual a razão para esta impossibilidade.

Foi depois apurado, com ajuda técnica de membros da ISA, de que este problema foi devido ao facto de se ter compilado o programa na arquitetura x64 do Linux da máquina virtual, quando a arquitetura do iMod é ARM9. No sentido de poder compilar código em C para a arquitetura de processador ARM9, recorreu-se a *cross-compiling*, que permite a compilação de código numa dada arquitetura estando noutra arquitetura, isto é, neste caso, compilar para ARM9 estando em x64. Para isso recorreu-se a *toolchains* (conjunto de ferramentas encadeadas).

Assim, qualquer programa em C que viesse a ser implementado poderia ser compilado num PC ou máquina virtual, e no futuro o mesmo poderia ser sempre compilado para ARM9 a fim de ser colocado no iMod.

4.7 RESULTADOS OBTIDOS NESTA FASE

Além de se poder conhecer bastante o dispositivo iMod e as suas potencialidades, no final desta fase obteve-se um conjunto de conhecimentos muito significativo, consistindo num leque que abrange muitos aspetos fulcrais do desenvolvimento de soluções de monitorização e controlo.

Infelizmente, algo muito importante contrariou as expetativas iniciais, que foi o facto de se descobrir que o iMod não permite compilar soluções desenvolvidas em C a fim de poder executar indo de encontro com a arquitetura de processador deste dispositivo, uma vez que a

Techbase retirou propositadamente esta funcionalidade para induzir a utilização da linguagem Java, para a qual é o alvo principal o desenvolvimento de soluções para o iMod.

Neste sentido, tem de se recorrer a *cross-compiling* para compilar soluções implementadas para a arquitetura ARM9 e se poder executar a solução compilada no iMod.

5. FASE DE REQUISITOS

Este capítulo tem como objetivo apresentar a especificação final da solução proposta. É efetuada uma análise, primeiramente em geral e depois em detalhe, sobre os requisitos do sistema total.

5.1 VISÃO GERAL

Decidida a integração do projeto na solução destinada a uma instituição bancária, pretendia-se implementar uma solução ISA aproveitando um *firmware* protocolar existente a ser remodelado (protocolo iHub), que no conjunto seria integrada no iMod.

Decorreram reuniões no âmbito da integração do projeto iMod na solução para a instituição bancária, com os seguintes objetivos:

- Perceber qual a abordagem que fora utilizada no âmbito do levantamento de requisitos face ao banco;
- Analisar as necessidades levantadas.

No seguimento dessas reuniões, foram fornecidos artefactos formais e informais envolvendo as necessidades do cliente e das propostas técnicas, tecnológicas, de equipamento, etc.

Esses artefactos consistiam num documento de pré-requisitos redigido pelo engenheiro responsável pelo levantamento de requisitos com o banco e alguns *e-mails* trocados no âmbito deste projeto, que definiam alguns detalhes importantes para o mesmo.

Foram lidos todos esses artefactos, tirando-se anotações e esclarecendo-se algumas dúvidas com os membros do projeto.

Após terem sido anotadas todas as necessidades que puderam ser retiradas dessa leitura, foi escrito um documento de requisitos de alto nível em conjunto com um estagiário agregado ao projeto corrente, sendo este documento frequentemente revisto pela coordenadora e responsáveis pelo projeto do banco.

Foi utilizado um histórico de versões, em que definíamos o documento como *baselined* quando submetido para revisão.

Continuou-se a melhorar o documento de requisitos de alto nível após o *feedback* obtido através de uma reunião de revisão deste mesmo documento.

Entretanto, houve mais uma reunião no âmbito do projeto de integração de iMod para o banco, com os seguintes objetivos:

- Revisões do documento de necessidades e de requisitos de alto nível do projeto;
- Discussão sobre passos seguintes.

Foi fornecida documentação sobre o protocolo iHub já existente; acesso a uma interface da ISA para um módulo que utiliza o protocolo iHub, que tem páginas muito semelhantes ao que se pretende fazer; indicações sobre campos a aparecerem ou desaparecerem dessa interface face à interface que é suposto ser implementada; acesso a código C do protocolo para se terem noções mais claras sobre aquilo em que consistia.

Criou-se um diagrama de Casos de Uso do sistema completo, assim como uma matriz de rastreabilidade Necessidades-Requisitos, os quais foram incluídos no documento de requisitos de alto nível, que foi progressivamente sofrendo melhorias (Anexo C).

5.2 CARACTERÍSTICAS DOS UTILIZADORES

Do ponto de vista de utilizadores do sistema, no âmbito de desenvolvimento no projeto de estágio, existem apenas aqueles que irão interagir com o iMod, ou seja, com a interface Web e respetivas configurações, e que neste caso são a pessoa ou pessoas responsáveis pela instalação e manutenção do iMod, e que se encontrarão em cada agência.

5.3 ANÁLISE DE REQUISITOS

5.3.1 Metodologia Usada na Recolha de Requisitos

A recolha de requisitos foi efetuada pela equipa responsável pelo projeto da instituição bancária, que se agregou à equipa de integração do iMod.

5.4 REQUISITOS FUNCIONAIS

5.4.1 Interface Web do iMod

Eis os requisitos de alto nível que foram definidos e que estão associados à interface Web do iMod (Anexo C - Especificação de Requisitos Alto Nível), em que apenas os 4 primeiros são obrigatórios/prioritários face aos restantes:

- Req01 – Acesso via *login* à interface do iMod;
- Req02 – Módulo *monitor*;
- Req03 – Módulo *server configuration*;
- Req04 – Módulo *channels configuration*;
- Req05 – Módulo *data explorer*;
- Req06 – Módulo *reports*;
- Req07 – Módulo *alarms*;
- Req08 – Exportar configurações;
- Req09 – Exportar *logs*.

Obtiveram-se mais detalhes sobre os requisitos, utilizando como referência uma interface Web já existente para o protocolo de recolha de dados (iHub), além de informação que era fornecida durante as reuniões a fim de serem indicadas diferenças relativamente à interface já existente (páginas correspondentes aos módulos *server configuration* e *channels configuration*).

Entretanto, surgiu a ideia de se criar uma página de configurações gerais, onde poderia ser efetuado o mapeamento entre valores de equipamentos vindos da canais de parâmetros configurados, a fim de ser mostrado o estado dos equipamentos na página de monitorização.

A interface Web do iMod tem de obter e configurar dados do protocolo de comunicação entre o iMod e o servidor central (Req10 - iHub).

		Matriz de Rastreabilidade Necessidades/Requisitos do projeto BAPOP																
		REQ01	REQ02	REQ03	REQ04	REQ05	REQ06	REQ07	REQ08	REQ09	REQ10	REQ11	REQ12	REQ13	REQ14	REQ15	REQ16	REQ17
Recolha de dados de consumo	NEC01																	
Controlar sistemas AVAC	NEC02																	
Controlo da iluminação interior da agência	NEC03																	
Controlo de iluminação dos anúncios/montras	NEC04																	
Ferramenta de manutenção local	NEC05	X	X	X	X	X	X	X	X	X								
Histórico de consumos	NEC06						X											
Emissão de relatórios	NEC07							X										
Emissão de configurações e logs	NEC08								X	X								
Protocolo de comunicação entre o iMod e o servidor central	NEC09										X							
Comunicação entre os aparelhos de medição e o iMod	NEC10											X						
Comunicação do iMod com o servidor central	NEC11												X					
Entradas e saídas digitais	NEC12													X	X			
Medição de temperatura	NEC13																X	
Permitir a opção de controlo manual sobre os circuitos	NEC14																X	

Tabela 1 - Matriz de Rastreabilidade de Necessidades/Requisitos

5.5 PROTÓTIPOS DE INTERFACE

Os protótipos de interface criados para a interface Web do iMod foram *mockups* que representavam as páginas Web consideradas aquando da definição de requisitos (Anexo D).

Esses *mockups* foram criados com base em toda a informação recolhida e com base na interface Web para leitura e configuração do protocolo iHub, anteriormente desenvolvido pela ISA.

Além do mais, tiveram-se em linha de conta algumas páginas Web da ISA (que não a do iHub) e o contexto do banco, na definição desses *mockups*.

5.6 REQUISITOS NÃO-FUNCIONAIS

5.6.1 Eficiência

- As soluções têm de ser construídas com uma arquitetura simples para permitir a sua execução de forma eficiente para o dispositivo iMod, que tem poucos recursos de processamento.

5.6.2 Fiabilidade

- As soluções têm de ser fiáveis ao ponto de informar o utilizador sobre o estado das operações efetuadas e garantindo que o sistema não seja interrompido.

5.6.3 Manutenção

- Tem de existir um ficheiro para configurações da interface Web do iMod.

5.6.4 Portabilidade

- A interface Web do iMod tem de ser implementada de modo a ser garantidamente compatível com o *browser* Google Chrome.

5.6.5 Segurança

- É necessário aplicar um algoritmo de encriptação na palavra-passe da conta de sessão de utilizador da interface Web do iMod.

5.6.6 Usabilidade

- A interface Web do iMod tem de ser compatível com *netbooks*, na medida em que a usabilidade e visibilidade sejam viáveis em resoluções reduzidas, como as dos ecrãs dos *netbooks*.

5.6.7 Requisitos Tecnológicos

- As soluções de *software* irão localizar-se na memória interna do dispositivo iMod.

5.7 ANÁLISE CRÍTICA

Os detalhes ao nível de requisitos foram ganhando pormenorização no decorrer do próprio desenvolvimento. Contudo, definidos os requisitos de alto nível e os *mockups*, já foi possível avançar para o desenho da arquitetura da solução e posteriormente iniciar a sua implementação.

6. ANÁLISE TECNOLÓGICA

Pretende-se analisar o estado do projeto e as tecnologias a usar na implementação do sistema.

6.1 ESTADO DO PROJETO

Após a fase de requisitos, seria desenvolvida uma interface Web para o iMod, que serviria para configurar e recolher valores do protocolo iHub alterado e a ser colocado no iMod e, no seguimento dessa informação, foi proposto que se optasse pela tecnologia/linguagem Web que permitisse maior conforto e/ou fosse mais adequada.

Consequentemente, investigou-se que tecnologia/linguagem deveria ser usada.

6.2 ANÁLISE DE TECNOLOGIAS PARA O SISTEMA

Quanto à decisão de como se iria fazer a interface Web, optou-se por se fazer em PHP, pois apesar de não se estar assim tão à vontade com desenvolvimento Web e do facto de nunca se ter aprendido PHP, a curva de aprendizagem é acessível e existem muitos tutoriais e documentação disponíveis na Internet.

Além disso, soube-se que não custaria correr uma solução em PHP no iMod, uma vez que já tem uma interface Web desenvolvida em PHP (NX Dynamics), assim como o servidor Apache para alocar, interpretar e correr as páginas.

6.3 FERRAMENTAS DE DESENVOLVIMENTO DE SOLUÇÕES EM PHP

Começou por se preparar um ambiente de desenvolvimento em Eclipse, mas este não se apresentou como a melhor forma de implementar em PHP, pois é muito confuso e tem falta de atalhos, entre outros fatores.

Na intenção de se aprender PHP, estar ambientado e preparar o ambiente de desenvolvimento para PHP, instalou-se o programa XAMPP, que contém vários utilitários no âmbito do desenvolvimento em PHP, tais como o Apache.

Após se ter tentado alguns editores para programar em PHP, acabou por ser sugerido o Sublime Text, que se apresentou bastante adequado, pois é um programa gratuito, reconhece esta linguagem e contém funcionalidades que facilitam a programação.

6.4 PHP NO IMOD

No iMod é a pasta “htdocs” que contém os projetos PHP para execução, mas carregada na memória RAM do iMod quando este é ligado, pelo que os projetos se mantêm gravados numa outra pasta, a partir da qual são copiados.

Para instalar a interface Web do iMod, tem de se a colocar na pasta persistente, a fim de ser carregada na inicialização.

6.5 APRENDIZAGEM DE PHP

Quando as noções de PHP foram minimamente adquiridas, procurou-se como fazer um cliente TCP em PHP, o que foi relativamente acessível. Continuou-se a experimentar PHP, seguindo tutoriais, já no intuito de estruturar a arquitetura da solução pretendida para o projeto.

6.6 CONCLUSÕES

A linguagem PHP é uma linguagem para aplicações Web que tem aspetos muito positivos, que são o facto de ter uma curva de aprendizagem muito elevada (nunca tendo utilizado esta linguagem anteriormente, é possível adquirir conhecimentos sobre esta muito rapidamente), ser bastante intuitiva, ser muito utilizada (logo, existem muitos tutoriais e documentação de qualidade na Internet) e de ser muito completa (tem muitas funcionalidades e bibliotecas, como por exemplo para o formato JSON, como se veio a descobrir e precisar mais tarde, após se ter começado a interface Web em PHP para o iMod).

7. FASE DE ARQUITETURA E IMPLEMENTAÇÃO

7.1 DEFINIÇÃO DE ARQUITETURA GERAL

Como se pode ver na Figura 19, foi definida a arquitetura geral das soluções no âmbito do projeto para a instituição bancária, em que é destacado o âmbito de desenvolvimento, que representa as soluções a serem desenvolvidas pelo autor.

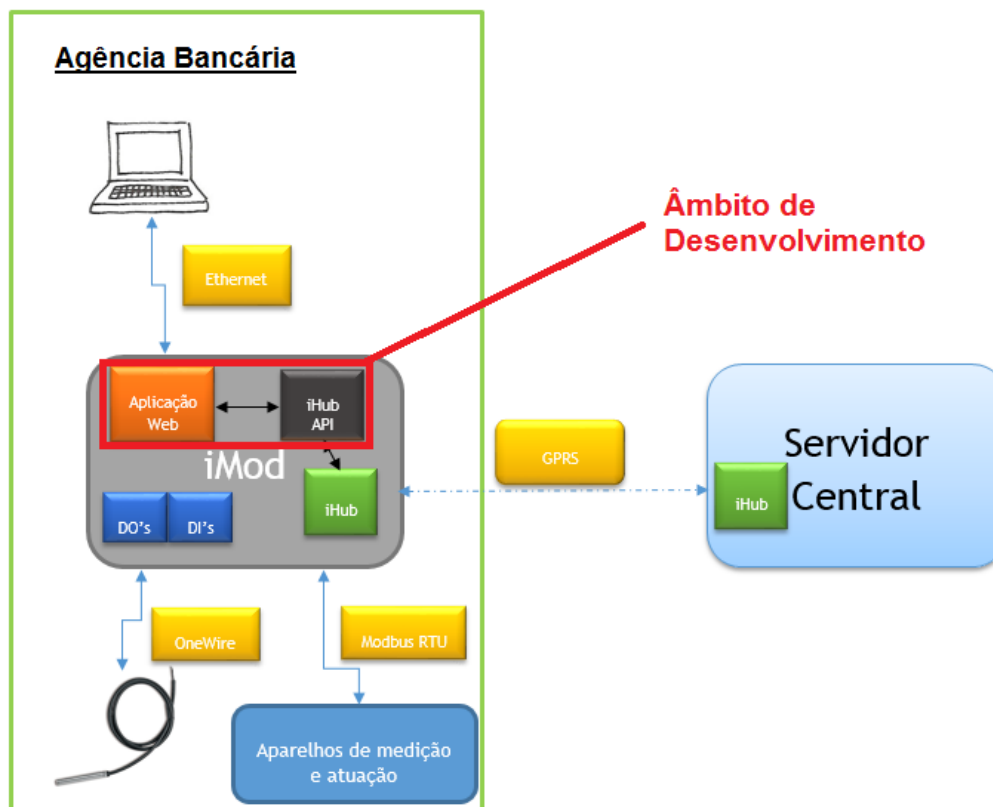


Figura 19 - Arquitetura geral com foco no âmbito de desenvolvimento

7.2 DEFINIÇÃO DE ARQUITETURA EM ÂMBITO DE IMPLEMENTAÇÃO

Tendo em foco o âmbito de implementação no projeto, foi definida a arquitetura para as soluções a serem implementadas: interface Web do iMod e API do iHub (será apresentada a razão para a sua necessidade).

É suposto utilizar-se o protocolo iHub, mas além deste ter um acesso que não é intuitivo, também não permite reconhecer o formato JSON, pelo que o acesso é feito com dados binários e não texto. Por conseguinte, propôs-se que deveria ser feita uma API (uma interface de aplicação) em C para fazer o elo de ligação entre a interface Web em PHP e o protocolo iHub, inseridos no iMod.

A interface Web anteriormente desenvolvida pela ISA para aceder ao iHub (como é o caso do iMeter) obtém e modifica diretamente as configurações do protocolo iHub, levando a uma maior dependência entre essas duas camadas de *software* (interface Web e protocolo iHub), impedindo a reutilização de aplicações em projetos e âmbitos diferentes da ISA. Esse é um problema que influencia muito a necessidade de se criar uma API para o protocolo iHub.

Chamou-se essa API de “iHub API”, pois seria precisamente uma interface para o protocolo iHub, que poderá eventualmente vir a representar um Webservice para o protocolo em questão, pois pretendia-se que se utilizassem *sockets* TCP/IP para esse efeito.

Após essa decisão, foi redigido um documento que consiste na sugestão da API para o protocolo iHub (Anexo F – Proposta de API do iHub), definindo os métodos existentes e as estruturas de pedidos e respostas inerentes às transações de acesso à API por parte de uma aplicação interessada, que é neste caso a interface Web para o iMod.

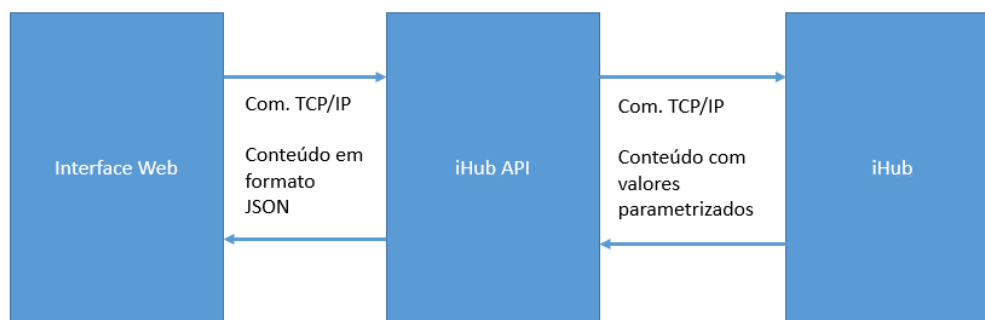


Figura 20 - Conjunto de soluções a introduzir no iMod.

Este conjunto de soluções é um contributo para o projeto para a instituição bancária.

Criou-se um diagrama para explicar a estrutura e interação da interface Web em PHP, sendo que este consiste num conjunto de diagramas divididos em 3 camadas de base no modelo *multi-tier*: camada de interface de utilizador; camada de lógica; camada de persistência e serviços (Anexo G).

7.3 IMPLEMENTAÇÃO

Tendo os *mockups* acabados e os ficheiros base em PHP prontos, utilizaram-se os *mockups* para construir a interface Web, aplicando os elementos HTML necessários às páginas estipuladas, com os devidos atributos.

No sentido de dar um aspeto agradável e boa usabilidade à interface Web, procuraram-se modelos de *stylesheet* CSS, sendo proposta uma muito boa, chamada Bootstrap.

Começou por se criar a página de Login que daria acesso às restantes páginas da aplicação através de autenticação com nome de utilizador e palavra-passe (único utilizador com credenciais definidas num ficheiro de configuração, com palavra-passe devidamente encriptada).

Utilizaram-se variáveis de sessão para conseguir criar a sessão de utilizador. Nesse contexto, foi de grande interesse aplicar um temporizador para a duração da sessão de utilizador, pelo que se utilizou JavaScript para visualizar um contador no ecrã, terminando a sessão de utilizador quando o tempo se esgotasse (duração de sessão é definida num ficheiro de configuração do *site*).

Criou-se, na interface Web para o iMod, uma página para configurações gerais, destinada a conter a configuração do mapeamento dos parâmetros de canais àquilo que se pretende que seja representado na página de monitorização de uma agência bancária. Esta página poderá conter outras configurações gerais no futuro.

Começaram por se desenvolver os métodos para a parte lógica, que consistiram em recolher e poder configurar os dados de configuração associados à página de configuração de servidor, e também a validação de campos de entradas, com respetivos limites e tipo de dados de entrada no HTML da respetiva página.

Após o *site* estar minimamente estruturado, principalmente para as páginas prioritárias (monitorização, configuração geral, configuração de servidor e configuração de canais), com a parte de função de lógica definidas, decidiu-se que se deveria utilizar o formato de dados JSON (e se necessário, ficheiros no formato JSON) pelo menos ao ponto de enviar e receber dados para o protocolo iHub.

O JSON é um formato de dados bastante usado, robusto e intuitivo para representação de dados, variáveis, objetos, etc., tanto no âmbito de comunicação, como de persistência. Para isso estudou-se o formato JSON, investigaram-se ferramentas nesse contexto e bibliotecas ou métodos para utilizar JSON em PHP e em C.

O PHP já contém métodos que codificam e descodificam informação no formato JSON, que são bastante fáceis de utilizar.

Quanto ao C, não existe nenhuma biblioteca para JSON que seja gratuita, fiável, simples e que contenha a possibilidade de estruturar dados por chave-valor. Nesse sentido pensou-se em optar por alternativas que possibilitassem ler e escrever JSON em C, que seria basicamente a de ler e construir as *strings* seguindo os padrões esperados.

Entretanto, desenvolveu-se código para conter configuração e leitura em JSON da configuração de portas série, entre outras informações relevantes para a interface de utilizador na interface Web em PHP.

Foi estruturado o código da interface Web de modo a emparelhar com a API do iHub que se tinha anteriormente definido.

No seguimento, fizeram-se melhorias gerais no interface Web do iMod; analisou-se o protocolo iHub; analisou-se e reestruturou-se o modelo da API; reescreveram-se e corrigiram-se pormenores da proposta da API; e continuou-se a estruturar o código PHP para emparelhar com a API do iHub.

Após todos esses aspetos definidos e elaborados, desenvolveu-se a base da API na linguagem C, no sistema operativo Linux Ubuntu.

Conseguiu-se, via *sockets* TCP/IP, ter contacto entre a interface Web e a API. A API está a ser um servidor TCP/IP à espera de pedidos no formato JSON, que correspondam aos pedidos definidos na proposta da API.

Sabendo qual a estrutura dos pedidos, a API facilmente conseguia ler o conteúdo JSON através de *parsing* do mesmo, e por conseguinte, enviar uma estrutura JSON como resposta.

Por outro lado, voltou-se a pegar na interface Web em PHP, na qual se implementou a parte de mapeamento de dados na página de configuração geral, a qual era persistida através de um ficheiro no formato JSON. Foi possível assim, mostrar-se na página de monitorização os valores vindos dos canais parametrizados para página de configuração de canais.

Para melhorar a qualidade do código e para permitir entendê-lo melhor para futura utilização, manutenção e continuação de desenvolvimento das aplicações implementadas, dedicou-se algum tempo a comentar todo o código e ficheiro de configuração que tinha sido elaborado.

Foram feitas melhorias tanto na API, como na interface Web, fazendo-se testes e executando-os, conseguindo-se assim obter a interface Web e a API iHub minimamente estáveis e consistentes.

Após essas melhorias, foi possível ter nesse momento a API a simular o protocolo iHub e a persistir as configurações em ficheiros, ou seja, ficheiros de texto contendo os valores das configurações.

Desenvolveram-se as funções da API iHub para obtenção e configuração do servidor e para a alteração de data e hora do servidor, ao ponto de simular a comunicação da API iHub com

o protocolo iHub, uma vez que ainda não se tinha o protocolo disponível, mas era conhecida a estrutura de pedidos e respostas para esse efeito.

Simulou-se o protocolo iHub recorrendo ao programa Hercules, em que num servidor TCP em escuta, se podia simular o envio de respostas pelo iHub, seguindo a estrutura da respetiva resposta, que está documentada num documento do iHub.

No final da fase de implementação, foram feitas correções nas soluções, de modo a poderem ser testadas para criação de versões finais das mesmas no âmbito do projeto de estágio.

7.3.1 Ferramentas

Para editar o código desenvolvido, usou-se o Sublime Text no Windows para a interface Web em PHP, e usou-se o Gedit no Linux para a API do iHub.

Para compilar e/ou executar o código desenvolvido, usou-se o Apache do XAMPP (no Windows em implementação; no Linux em integração) para a interface Web em PHP, e usou-se o GCC no Linux para a API do iHub.

7.3.2 Organização do Código

7.3.2.1 Interface Web do iMod

- Raiz do projeto: páginas Web PHP;
- Pasta “methods”: ficheiros PHP que contêm a parte de lógicas da página (métodos, classes, ...);
- Pasta “conf”: para ficheiro de configuração do interface Web do iMod;
- Pasta “css”: contém os ficheiros de *stylesheet* CSS;
- Pasta “fonts”: contém ficheiros de fonte para texto;
- Pasta “helpers”: componentes genéricas para as páginas PHP, tais como o cabeçalho e o rodapé;
- Pasta “images”: contém as imagens a serem utilizadas na aplicação Web, como por exemplo o logotipo que se encontra no cabeçalho das páginas Web;
- Pasta “js”: contém os ficheiros JavaScript;
- Pasta “json_data”: contém ficheiros JSON com configurações locais, como é o caso do mapeamento de parâmetros para os equipamentos.

7.3.2.2 API do iHub

A API do iHub é desenvolvida na linguagem C e encontra-se representada através de 2 ficheiros: “ihubapi.h”; “ihubapi.c”.

7.4 COMPONENTES

7.4.1 Interface Web do iMod

Na Figura 21 é possível ver as componentes da camada de interface de utilizador da interface Web do iMod através de um diagrama representativo das várias componentes da camada de interface de utilizador e sua respetivas interações.

Ao aceder à raiz do projeto no Apache, este dirige-se sempre ao “index.php”, pelo que aparece logo como ponto inicial no diagrama, mas é logo redirecionado para a página “login.php” (página de Login), que por sua vez é redirecionado para a página “monitor.php” (página Monitor, que é a página “home” da sessão de utilizador), caso a sessão de utilizador já se encontre inicializada, ou no caso de se efetuar *login* com sucesso através da página para Login.

As páginas têm componentes que lhes conferem aspetos em comum, como é o caso do cabeçalho (“header.php”), do rodapé (“footer.php”) e duma componente para as mensagens de informação ao utilizador (“action.php”), tais como após ter efetuado uma configuração, para saber se foi efetuada com sucesso (a página de Login não tem esta última componente, apesar de ter mensagens no seu contexto).

As páginas Web que representam os módulos do programa são as seguintes: “monitor.php” (página Monitor); “serverconf.php” (página Server Configuration); “channelsconf.php” (página Channels Configuration); “general conf.php” (página General Configuration); e as restantes, que são um conceito a longo prazo, sendo estas “dataexplorer.php” (página Data Explorer), “reports.php” (página Reports) e “alarms.php” (página Alarms).

Existem ainda as componentes “logout.php”, que termina a sessão de utilizador e redireciona para a página de Login; e “resetcountdown.php”, que reinicializa o tempo de sessão de utilizador na contagem decrescente em questão.

Para mais informação sobre as componentes da interface Web do iMod, mais concretamente em relação às duas outras camadas (camada de lógica e camada de persistência e serviço), ver o Anexo G - Arquitetura de interface Web do iMod.

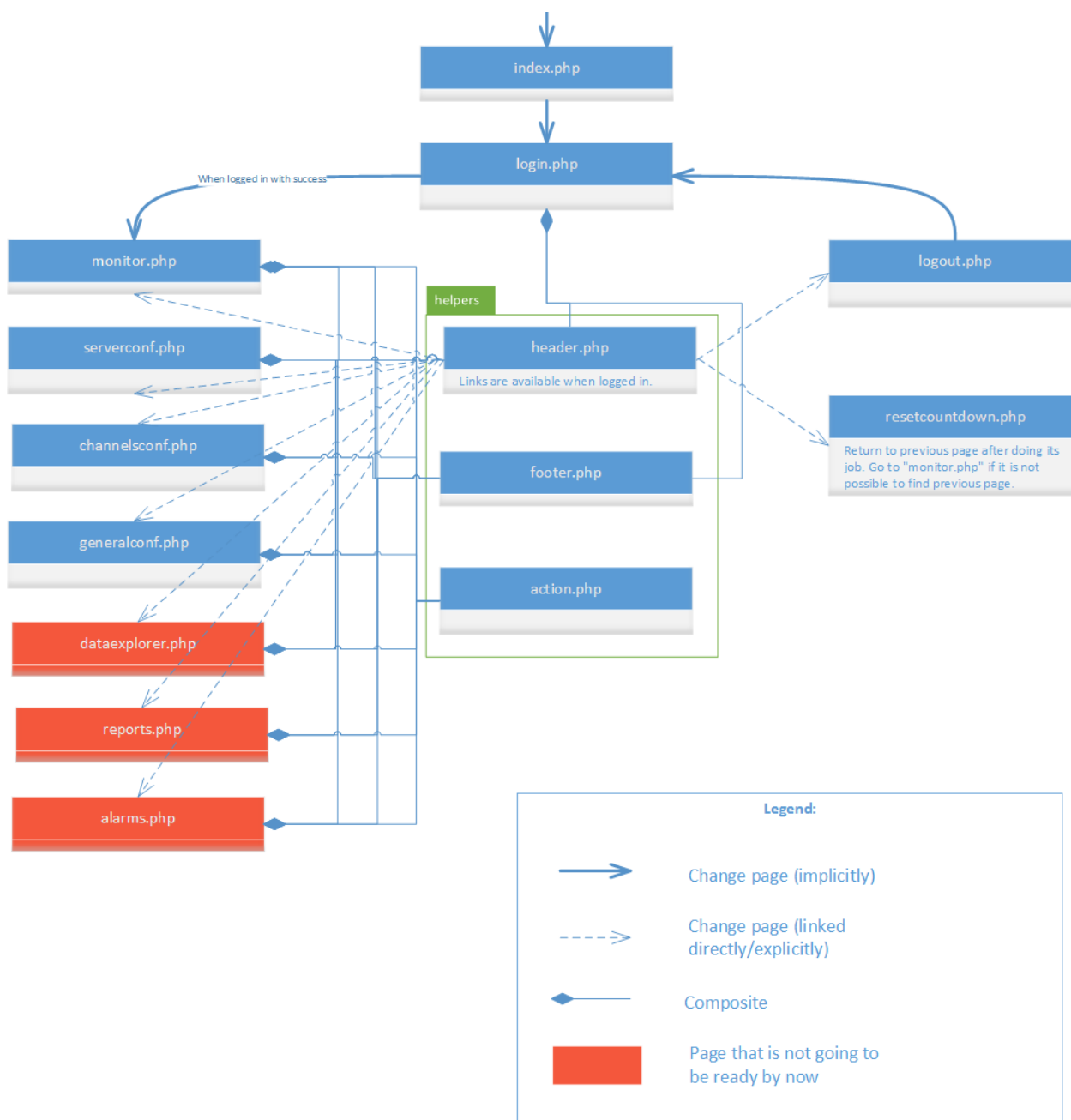


Figura 21 - Diagrama representativo das várias componentes da camada de interface de utilizador e suas respectivas interações (ver legendas do diagrama).

7.4.2 API do iHub

Implementou-se uma API para o iHub, em que foi inicialmente estruturada a sua arquitetura através de um documento de proposta da API em questão (para mais informação, ver Anexo F - Proposta de API do iHub).

A arquitetura apresentada consiste na definição das estruturas de pedidos e respostas no formato JSON inerentes às respetivas transações a efetuar (para obter e configurar dados do protocolo iHub). Eis os códigos dos pedidos e respostas que fazem parte desta API:

7.4.2.1 Request (pedido)

Aqui são apresentadas as combinações de código-função que serão associadas a um pedido por parte da interface Web.

ID Função

1. `server_conf getServerConf();`//Obtém configuração de servidor remoto associado ao iHub. Devolve uma estrutura `server_conf`.
2. `int setServerConfTime(unsigned int year, unsigned int month, unsigned int day, unsigned int hours, unsigned int minutes, unsigned int seconds);`//Configura uma nova data no servidor remoto associado ao iHub. Poderá usar “struct `server_conf_time`” no lugar como argumento. Retorna valor correspondente à forma como decorreu a operação.
3. `int connectServer(char[] host, unsigned int port);`//Testa conexão a um *host* remoto e porto. Retorna valor correspondente à forma como decorreu a operação.
4. `int setServerConf(unsigned int serial, unsigned int as_tcp_server, char host[], unsigned int port, unsigned int comm_period, unsigned int encap_crc, unsigned int errors_before_reboot);`//Configura dados principais de configuração de servidor remoto associado ao iHub. Poderá usar “struct `server_conf_server`” no lugar como argumento. Retorna valor correspondente à forma como decorreu a operação.
5. `channel_params_conf getChannelsConf();`//Obtém configuração de canais de parâmetros. Devolve uma estrutura `channel_params_conf`.
6. `int setChannelsConf(channel_param_conf params[]);`//Configura canais de parâmetros. Retorna valor correspondente à forma como decorreu a operação.
7. `int setChannelConf(channel_param_conf param, unsigned int index);` //Configura um canal de parâmetro com um índice especificado. Retorna valor correspondente à forma como decorreu a operação.

7.4.2.2 Response (resposta)

Aqui são apresentados os códigos de retorno que serão associados a uma resposta para a interface Web.

ID Representação

0. `SUCCESS_RESPONSE` - Sucesso

1. JSON_FAIL_RESPONSE - Má formatação de formato JSON de pedido ou de resposta (e por não corresponder aos valores, tipos de valores e/ou estrutura esperada segundo a API).
2. FAIL_RESPONSE - Falha da API ou operação apresenta resultado negativo (por exemplo, o teste de conexão a servidor falhou).
3. TECH_ISSUE_RESPONSE – Falha técnica (por exemplo, não conseguiu criar um *socket* para ligar a servidor no teste de conexão).

7.5 DESIGN DE INTERFACE

No sentido de dar um aspeto agradável e utilização intuitiva à interface Web, usaram-se modelos de *stylesheet* CSS, nomeadamente o Bootstrap, atrás referidos.

7.6 ANÁLISE CRÍTICA

As diversas aplicações não ficaram terminadas a ponto de poderem ser instaladas nas agências bancárias, mas a sua base é a prova de que é possível integrá-las no dispositivo iMod.

É ainda necessário testá-las, melhorá-las e corrigir possíveis falhas encontradas, a fim de garantir a qualidade e confiabilidade desejáveis no funcionamento das mesmas.

8. FASE DE VERIFICAÇÃO E VALIDAÇÃO

8.1 PLANO DE TESTES

É necessário criar um plano de testes e executá-lo, a fim de garantir a qualidade e confiabilidade desejáveis no funcionamento das soluções implementadas.

Para esse efeito, fez-se o plano de teste em Excel através de um *template* fornecido pela ISA para esse propósito (Anexo E - Plano de Testes da interface Web do iMod).

Foi considerado a implementação de testes unitários para as soluções, pois permite cobrir e executar as mesmas de forma rápida e eficiente, no entanto, não houve oportunidade de explorar e efetuar esses mesmo testes.

Criou-se uma primeira versão do plano de testes quando já se tinha um protótipo da interface Web do iMod.

8.1.1 Elaboração do plano de testes

Começou-se por escrever um plano de testes que incluísse a parte de sessão de utilizador (*login*, *logout*, temporizador de sessão, etc.), a parte de configuração de servidor e da configuração de parâmetros de canais.

Foram elaboradas 3 versões do plano de testes, que foram seguidas individualmente por uma execução e a criação de versões das aplicações: inicialmente apenas para a interface Web do iMod, e entretanto, já incluindo a API do iHub.

8.1.2 Execução do plano de testes

Numa primeira execução, o plano de testes foi executado por outra pessoa, a fim de se poder fechar uma versão da interface Web do iMod, respeitando a nomenclatura da ISA para versões de aplicações.

A pessoa que executou o plano de testes foi um estagiário de engenharia eletromecânica atribuído ao projeto de integração do iMod.

Por sua vez, o autor deste relatório pôde executar o plano de testes do colega para o código por ele implementado (aplicação de comunicação por porta série no Linux, desenvolvida em C, como contributo para o protocolo iHub).

As próximas execuções do plano de testes seriam efetuadas pelo autor, que foram intercaladas com correções. Na última execução, foi possível ter-se uma versão final das soluções de *software* no âmbito do projeto de estágio, em que todos os testes passaram com sucesso.

Preparou-se um computador *desktop* com o Linux Ubuntu de modo a permitir a integração da interface Web do iMod e da API iHub em C, para elaborar um ambiente de simulação e permitir desenvolvimento de funcionalidades de mais baixo nível de forma mais acessível, garantido acesso mais direto ao *hardware* em uso (Ex.: PLC por porta série).

Foi possível criar uma prova de conceito da comunicação com o protocolo iHub, que se comprovou ser um sucesso, viabilizando uma vez mais a integração das soluções (interface Web e API) no iMod, que corresponde às expectativas do projeto de integração de soluções no iMod.

Entretanto, tentou-se encontrar uma forma de se obter o protocolo iHub para testar a API com o mesmo, mesmo não sendo a versão para a instituição bancária, mas neste caso a versão que estava definida no documento que contém a definição do protocolo (do dispositivo iMeter).

Após discussão com os responsáveis pelo desenvolvimento do protocolo iHub, achou-se uma forma prática e viável de testar a API com o protocolo iHub: através da comunicação com um dispositivo iMeter, que contém o protocolo iHub implementado para este dispositivo.

Através de um programa da ISA para descobrir dispositivos na rede ou através da linha de comando utilizando o comando “arp -a” (através do MAC *address* que identifica o dispositivo), é possível descobrir o IP para aceder ao iMeter.

Ao aceder à interface Web do iMeter, sabendo o seu IP, é possível configurar a porta de escuta, e conseqüentemente, configurar a API (IP e porto) para poder aceder ao protocolo iHub inerente ao iMeter.

Após ter o dispositivo iMeter ligado por cabo Ethernet a um *switch*, efetuar a configurações em questão, e executar a API do iHub e o Apache para aceder à interface Web para o iMod (no computador com sistema operativo Linux instalado), foi possível visualizar e configurar com sucesso a data do protocolo iHub através da interface Web para o iMod.

Mais tarde, de forma a criar um ambiente de teste de integração o mais aproximado do real (agência bancária), montou-se um ambiente de simulação, no intuito de recriar as soluções de *software* no iMod.

Para isso, configurou-se um *router* para se poder ligar facilmente os respetivos dispositivos: *netbook* para aceder à interface Web do iMod; PC com Linux, contendo a interface Web do iMod e a API do iHub; e o dispositivo iMeter, que contém o protocolo iHub (ver Figura 22).

Através de DHCP, neste caso DHCP estático, puderam atribuir-se e IP's fixos para os endereços MAC dos respetivos, pois assim é possível ligar apenas os dispositivos ao *router*, garantindo que estes tenham os IP's configurados, e assim permitir facilmente criar o ambiente de simulação.

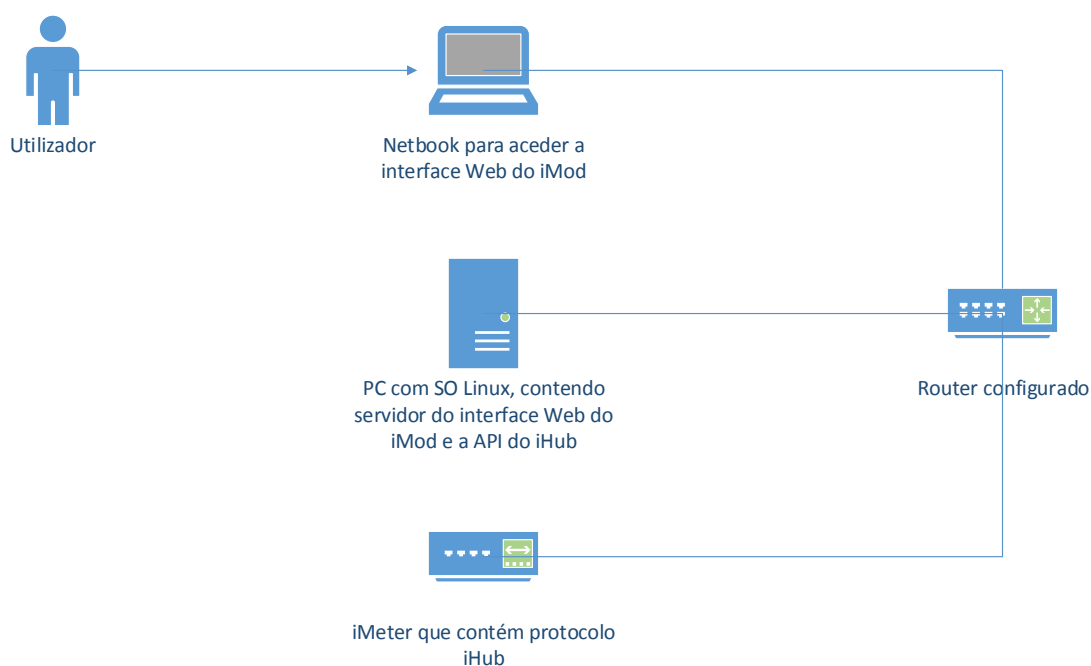


Figura 22 - Ambiente de teste de integração

Conseguiram integrar-se nessa máquina ambas as soluções com sucesso.

8.2 REVISÃO

Não foram efetuadas revisões às soluções implementadas, no entanto tinha sido proposta uma tarefa de revisão de código pela coordenadora, em que esta iria executá-la, mas não surgiu oportunidade desta tarefa ter sido realizada por falta de tempo.

8.3 VALIDAÇÃO

A validação foi efetuada pelas execuções do plano de testes.

8.4 VERIFICAÇÃO

Verificou-se que os requisitos prioritários foram implementados com sucesso, utilizando-se para isso a matriz de Necessidades/Requisitos criada durante a fase de requisitos. Essa verificação foi feita em conjunto com a restante equipa do projeto da instituição bancária.

8.5 INSTALAÇÃO NO CLIENTE

Não houve instalação no cliente, uma vez que o sistema ainda não se encontra completo e operacional para esse efeito.

No entanto, foi elaborado um manual para a interface Web do iMod em relação à última versão implementada (Anexo H - Manual de utilizador da interface Web do iMod).

9. CONCLUSÃO E PERSPETIVAS FUTURAS

9.1 PLANEAMENTO EFECTIVAMENTE SEGUIDO

Tentou-se levar a bom o planeamento inicial, que foi seguido da melhor forma possível, apesar de a solução não estar finalizada para ser colocada em campo. Não surpreende, dado que o projeto (re)surge num âmbito redefinido a meio do estágio, consistindo na integração do iMod no projeto associado à instituição bancária.

9.1.1 Justificação dos Desvios

Houve muitos fatores e decisões que influenciaram o decorrer do estágio, mas crê-se que sejam perfeitamente naturais em meios empresariais: mudança de instalações; decisão de propostas e contrato com clientes; o facto das pessoas na empresa estarem atribuídas a mais do que um projeto, não podendo estar sempre disponíveis quando necessário; entre outros fatores.

Certos aspetos podem reforçar algumas competências, tais como a adaptação, por causa de se ser mais autónomo/autodidata, tomar alguma iniciativa, ter maior motivação, organizar-se melhor e melhorar a comunicação no meio empresarial.

9.2 AVALIAÇÃO DO TRABALHO DESENVOLVIDO

Apesar das soluções não estarem efetivamente operacionais, para aplicar à instituição bancária, a base destas está implementada, bem testada e funcional, sendo que é garantida a execução destas com grande margem de sucesso no dispositivo iMod.

Além do mais, estas soluções (mais concretamente a interface Web do iMod e a API do iHub) poderão ser usadas futuramente em vários dispositivos e em vários âmbitos/projetos da ISA.

9.3 TRABALHO FUTURO

Como trabalho futuro surgem de imediato os módulos Data Explorer, Reports e Alarms da interface Web do iMod, assim como melhorias de usabilidade, entre outros fatores.

9.4 CONSIDERAÇÕES PESSOAIS

Nestes quase 9 meses de estágio, foi possível aprender bastante, contactar com um ambiente profissional e indiretamente com um cliente real (instituição bancária), pelo que foi

muito gratificante, tanto ao nível de conhecimentos e de experiência profissional, como de interação interpessoal.

10. REFERÊNCIAS BIBLIOGRÁFICAS

1. ISA (2013). <http://www.isasensing.com/pt/>. ISA – INTELLIGENT SENSING ANYWHERE (página internet oficial), Coimbra. (consultado em 09-09-2014)
2. TECHBASE. <http://www.techbase.eu/en/>. TECHBASE (página internet oficial), Polónia (consultado em 09-09-2014)
3. TECHBASE. <http://www.techbase.eu/en/produkty/imod-modul-telemetryczny.html>. iMod (consultado em 09-09-2014)
4. TECHBASE. <http://www.techbase.eu/en/rozwiazanie/systemy-powiadamiania.html>. NPE (consultado em 09-09-2014)
5. TECHBASE. <http://www.techbase.eu/en/rozwiazania/dedykowany-interfejs-graficzny.html>. NX Dynamics (consultado em 09-09-2014)
6. <http://json.org/>. Introducing JSON (consultado em 09-09-2014)
7. Wikipedia. <http://pt.wikipedia.org/wiki/Modbus>. Modbus (consultado em 09-09-2014)
8. Wikipedia. http://en.wikipedia.org/wiki/General_Packet_Radio_Service. GPRS (consultado em 19-09-2014)
9. Wikipedia. <http://pt.wikipedia.org/wiki/GSM>. GSM (consultado em 19-09-2014)
10. Computer Hope. <http://www.computerhope.com/atcom.htm>. AT commands (consultado em 19-09-2014)
11. Wikipedia. <http://en.wikipedia.org/wiki/SCADA>. SCADA (consultado em 19-09-2014)
12. Movicon. <http://www.movicon.com.au/>. Movicon (página internet oficial), Austrália (consultado em 19-09-2014)
13. Scrum. <https://www.scrum.org/>. Scrum (página internet oficial) (consultado em 19-09-2014)
14. HW group. http://www.hw-group.com/products/hercules/index_en.html. Hercules (página internet oficial) (consultado em 20-09-2014)

15. Sublime Text. <http://www.sublimetext.com/>. Sublime Text (página internet oficial) (consultado em 20-09-2014)
16. Apache Friends. https://www.apachefriends.org/pt_br/index.html. XAMPP (página internet oficial) (consultado em 20-09-2014)
17. Ubuntu. <http://www.ubuntu.com/>. Linux Ubuntu (página internet oficial) (consultado em 20-09-2014)
18. PHP. <http://php.net/>. PHP (página internet oficial) (consultado em 20-09-2014)
19. W3Schools. <http://www.w3schools.com/php/>. PHP Tutorial (consultado em 20-09-2014)
20. Atlassian. <https://www.atlassian.com/software/jira>. JIRA (página internet oficial) (consultado em 20-09-2014)
21. GCC. <https://gcc.gnu.org/>. GCC (página internet oficial) (consultado em 20-09-2014)
22. Apache Subversion. <https://subversion.apache.org/>. SVN (página internet oficial) (consultado em 20-09-2014)
23. Bootstrap. <http://getbootstrap.com/>. Bootstrap (página internet oficial) (consultado em 20-09-2014)
24. W3Schools. <http://www.w3schools.com/js/>. JavaScript Tutorial (consultado em 20-09-2014)
25. Balsamiq. <https://balsamiq.com/>. Balsamiq Mockups (página internet oficial) (consultado em 20-09-2014)
26. Sparx Systems. <http://www.sparxsystems.com/products/ea/>. Enterprise Architect (página internet oficial) (consultado em 20-09-2014)
27. Microsoft Office. <http://office.microsoft.com/en-us/visio/>. Visio (página internet oficial) (consultado em 20-09-2014)
28. Eclipse. <https://www.eclipse.org/>. Eclipse (página internet oficial) (consultado em 20-09-2014)
29. ARM. <http://www.arm.com/>. ARM (página internet oficial) (consultado em 20-09-2014)

30. Wikipedia. http://en.wikipedia.org/wiki/Programmable_logic_controller. PLC (consultado em 20-09-2014)
31. Wikipedia. http://pt.wikipedia.org/wiki/Machine_to_Machine. M2M (consultado em 20-09-2014)
32. Wikipedia. <http://pt.wikipedia.org/wiki/API>. API (consultado em 20-09-2014)
33. WISE-FTP. <http://www.wise-ftp.com/>. WISE-FTP (página internet oficial) (consultado em 20-09-2014)
34. WinSCP. <http://winscp.net/eng/download.php>. WinSCP (página internet oficial) (consultado em 20-09-2014)
35. Notepad++. <http://notepad-plus-plus.org/>. Notepad++ (página internet oficial) (consultado em 20-09-2014)
36. GNOME. <https://wiki.gnome.org/Apps/Gedit>. Gedit (página internet oficial) (consultado em 20-09-2014)
37. Application note on SCADA / Process Automation system. <http://www.sitech-bitdriver.com/tech/drawings/scada.pdf>. 2012 S.I. Tech, Inc. Application Note on SCADA (consultado em 20-10-2014)

11. ANEXOS

A. Anexo A - Proposta de Estágio

Proposta de Projeto (Engenharia Informática e Sistemas) Ano Letivo de 2013/2014	
Título de Projeto: ISA, iMod and ISA platform Integration	A quantos alunos se destina: 1
<p>Perfil de especialização dos alunos: Engenharia informática e sistemas, programação em C#, SQL e integração de sistemas.</p>	
<p>Contextualização: A ISA é uma empresa de referência no mercado da telemetria, o <i>core business</i> é a telemetria de tanques de GPL industriais, para além destas aplicações a ISA disponibiliza um leque variado de soluções de telemetria que permitem a gestão inteligente de frotas, leitura automática de contadores e a eficiência energética (http://www.isasensing.com/index.php?section=home).</p> <p>No âmbito da monitorização e controlo de redes de abastecimento de água e iluminação pública a ISA tem o objetivo de integrar nas suas plataformas equipamentos de terceiros, tanto controladores como dispositivos periféricos.</p>	
<p>Objetivos do trabalho:</p> <p>O objetivo passa por integrar o controlador iMod da Techbase nas plataformas ISA. Este equipamento tem várias funcionalidades essenciais para a ISA, tais como datalogging, funcionamento em modo PLC, comunicação através de GSM/GPRS, controlo de dispositivos periféricos, entre outros. Todas as funcionalidades do iMod podem ser acedidas e reprogramadas a partir do sistema operativo Linux.</p> <p>Para a integração entre o iMod e as plataformas ISA é necessário conhecer o funcionamento do iMod assim como as plataformas ISA, de forma a escolher a estratégia que melhor se adapte à integração dos dois sistemas. O desenvolvimento de um módulo que monitorize o sistema é essencial e parte integrante deste trabalho.</p>	
<p>Plano de trabalhos:</p> <p>Desenho da integração do iMod nas plataformas ISA será dividido nos seguintes passos:</p> <ol style="list-style-type: none"> 1. Estudo do estado da arte sobre integração de equipamentos em sistemas SCADAS e OPC. Estudo das plataformas ISA e análise das funcionalidades do controlador iMod; 2. Análise de requisitos e desenho da arquitetura do sistema e modelos de dados para os principais componentes do sistema, assim como mecanismos de deteção de falhas do sistema. 3. Análise e desenvolvimento do protocolo de comunicação da plataforma ISA com o iMod. 4. Desenvolvimento do módulo de monitorização e deteção de falhas do sistema. 5. Testes à solução desenvolvida. 	

6. Documentação.

Cronograma de trabalho:

Tarefas	N	N+1	N+2	N+3	N+4	N+5	N+7	N+8
T1	█							
T2		█						
T3			█					
T4				█	█	█		
T5							█	
T6								█
Metas		M1 M2	M3				M4	M5

Metas:

- M1- apresentação com resultados do trabalho;
- M2- documentação com a arquitetura do sistema;
- M3- apresentação do trabalho desenvolvido e definição do plano para as fases de desenvolvimento e testes;
- M4- teste à solução desenvolvida;
- M5- relatório final de estágio;

Orientador responsável:

David Macário
dmacario@isa.pt
 Rita Carreira
rcarreira@isa.pt

Local do Projeto:

O desenvolvimento do projeto decorrerá nas instalações da ISA - Intelligent Sensing Anywhere.

ISA - Intelligent Sensing Anywhere, S.A.
 Estádio Cidade de Coimbra, 92
 3030-320 Coimbra

Entidade acolhedora: ISA - Intelligent Sensing Anywhere

Supervisor na entidade acolhedora:

Andreia Carreiro (acarreiro@isa.pt)

B. Anexo B - Procura de Biblioteca Modbus-TCP

Integração do iMod

À procura de uma biblioteca para comunicação com protocolo Modbus TCP

1 Introdução

No âmbito do projeto de integração do iMod, pretende-se implementar comunicação que recorra ao protocolo Modbus TCP, em que essa comunicação fará ponte entre a aplicação cliente (de controlo) e o dispositivo iMod (que representa o servidor).

Nesse sentido, decidiu-se procurar uma biblioteca de Modbus TCP de modo a facilitar e acelerar a implementação da lógica inerente ao protocolo e respetivas relações.

O desenvolvimento do código é realizado no IDE Eclipse na linguagem Java, pelo que a biblioteca tem de ser compatível com estes aspetos.

Após encontrar uma biblioteca e contextualizar-se com a mesma é importante criar um projeto experimental de modo a garantir o bom funcionamento, entender melhor o potencial da biblioteca e saber instalá-lo para execução.

2. Biblioteca para Modbus TCP

2.1 Pesquisa de biblioteca Modbus TCP

Não foi preciso procurar muito para encontrar a biblioteca pretendida, pelo que a biblioteca em questão é designada de “jamod”, que é gratuita e desenvolvida em Java.

Esta biblioteca é adequada à nossa necessidade pois é compatível com o nosso ambiente de desenvolvimento, é gratuita, tem documentação associada (na Internet), permite acesso ao código fonte da mesma (na Internet) e é bastante utilizada (onde se poderão encontrar exemplos, dúvidas em fóruns, etc. frequentemente e é muito sugerida).

A biblioteca “jamod” encontra-se disponível para *download* no *link* seguinte: <http://sourceforge.net/projects/jamod/>

A respetiva **documentação** encontra-se neste *link*: <http://jamod.sourceforge.net/>

Encontrei um bom repositório onde é possível encontrar as classes da biblioteca e ver o respetivo **código fonte**. Eis o *link* para o repositório em questão: <http://grepcode.com/project/repo1.maven.org/maven2/net.wimpi/jamod/>

Nota: Existem bibliotecas ASP para comunicação Modbus TCP e que também são muito utilizadas, mas estas não se encaixam de todo com o ambiente de desenvolvimento e tem de se pagar licença da Microsoft para o efeito de utilização.

2.2 Apresentação genérica de biblioteca encontrada

A biblioteca “jamod” permite 3 diferente tipo de comunicação Modbus: TCP, UDP e Serial; das quais utilizaremos a de Modbus TCP.

O objetivo é existirem 2 módulos/classes que comunicam entre eles enviando pacotes de mensagem respeitando o protocolo Modbus TCP, neste caso. Um é *Master* (mestre), que tem função de cliente e que deverá se encontrar numa aplicação de controlo; o outro é o *Slave* (escravo), que tem função de servidor e que deverá se encontrar no iMod a fim de atender os pedidos de vários clientes (Figura 23).

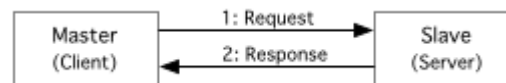


Figura 23 – Transação *Master-Slave*

A comunicação é para ser feita via Ethernet utilizando o protocolo TCP/IP, que está incluído no protocolo Modbus TCP, porque o protocolo TCP/IP encapsula a mensagem de protocolo Modbus, seja ASCII ou RTU (Figura 24).

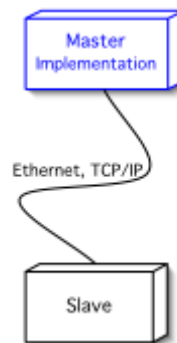


Figura 24 – Representação de uma simples ligação dos 2 módulos

Aqui se encontram todos os **documentos que detalham os interfaces e classes da biblioteca**, tendo em conta as possíveis relações de dependências/herança: <http://jamod.sourceforge.net/apidocs/index.html>

Existem **constantes com valores definidos** na biblioteca em que estes estão referidos neste *link*: <http://jamod.sourceforge.net/apidocs/constant-values.html>

Essas constantes fazem parte do interface designado Modbus, podendo se encontrar os detalhes sobre o mesmo e que são de grande interesse: <http://jamod.sourceforge.net/apidocs/net/wimpi/modbus/Modbus.html>

É ainda possível encontrar todas as **classes, interfaces e métodos listados e classificados alfabeticamente**, acompanhados de uma breve descrição e sendo possível aceder a mais detalhes clicando para o efeito. Eis o *link* em questão: <http://jamod.sourceforge.net/api/index-all.html>

Existe um termo utilizado na biblioteca que é “**process image**” (**imagem de processo**), que é a entidade que representa o conjunto de parâmetros de valores utilizados na comunicação Modbus e que podem representar valores provenientes das máquinas (para uso monitorização e controlo remoto). Mais informação em <http://jamod.sourceforge.net/kb/processimage.html>

2 Implementação de comunicação usando Modbus TCP

2.1 Apresentação detalhada da biblioteca “jamod”

Apresentam-se algumas das principais classes a utilizarmos, em que umas são obrigatórias para o caso e que outras são facultativas consoante a necessidade. Desta forma poder-se-á estar mais contextualizado para ao abordar a parte de implementação do código.

Existe antes de mais a interface **Modbus**, como referido mais acima, que define todas as constantes relacionadas com o protocolo Modbus.

Para o lado do *Slave* (servidor):

- **ModbusTCPListener:** Se estiver em escuta, aceita pedidos chegantes tratando-os para serem manipulados.
- **SimpleProcessImage:** Classe que implementa uma imagem de processo simples para ser capaz de executar testes unitários ou lidar com casos simples.
- **Tipo de parâmetros existentes para adicionar à imagem de processo:**
 - **Digital Input**(para *discrete input*) – Ex: SimpleDigitalIn
 - **Digital Output** (para *discrete output* ou *coil*) – Ex: SimpleDigitalOut
 - **Input Register** (para *input register*) – Ex: SimpleInputRegister
 - **Register** (para *holding register*) – Ex: SimpleRegister
- **ModbusCoupler:** Classe implementada seguindo um padrão Singleton (só pode ter uma instância da classe), para acoplar o escravo com um mestre ou com um dispositivo. De momento, este apenas fornece uma referência ao modelo OO (Orientado a Objetos) da imagem de processo.

Para o lado do *Master* (cliente):

- **TCPMasterConnection:** Constrói uma instância TCPMasterConnection com um determinado endereço de destino.
- **ModbusTCPTransaction:** Classe que implementa a interface ModbusTransaction. Uma operação é definida pela sequência de envio de uma mensagem de pedido e receber uma mensagem de resposta relacionada.

- **ModbusMessageImpl:** Classe abstrata implementando um ModbusMessage. Esta classe oferece implementações especializadas com a funcionalidade que eles têm em comum. Representa a mensagem a ser utilizada na transação.
 - **ModbusRequest:** Classe abstrata implementando um ModbusRequest. Esta classe oferece implementações especializadas com a funcionalidade que eles têm em comum. Contém lógica de pedido da mensagem.
 - **ReadInputDiscretesRequest:** Classe que implementa um ReadInputDiscretesRequest. A implementação correlaciona-se diretamente com a função de leitura de **Input Discretes**. Ela encapsula a mensagem de pedido correspondente. Input Discretes são entendidos como bits que não podem ser manipulados (ex: ativado ou desativado).
 - **E outras mais para pedido:** De leitura ou escrita para os vários tipos de parâmetros.
 - **ModbusResponse:** Classe abstrata implementando um ModbusResponse. Esta classe oferece implementações especializadas com a funcionalidade que eles têm em comum. Contém lógica de resposta da mensagem.
 - **ReadInputDiscretesResponse:** Classe que implementa um ReadInputDiscretesResponse. A implementação correlaciona-se diretamente com a função de leitura de **Input Discretes**. Ela encapsula a mensagem de pedido correspondente. Inputs Discretes são entendidos como bits que não podem ser manipulados (ex.: ativado ou desativado).
 - **E outras mais para resposta:** De leitura ou escrita para os vários tipos de parâmetros.

2.2 Implementação de projeto de teste

Para experimentar o funcionamento da biblioteca, decidiu-se fazer um projeto em Java para esse efeito, neste caso, 2 projetos Java, em que um é o Master (cliente) e outro é o Slave (server).

Os projetos foram desenvolvidos com base nos exemplos que se encontram na página oficial de biblioteca e que contém documentação, pelo que puderam ser feitas alterações mínimas a título de experiência.

A parte de implementação associada ao Master encontra-se neste *link*: http://jamod.sourceforge.net/kb/tcp_master_howto.html

A parte de implementação associada ao *Slave* encontra-se neste *link*:
http://jamod.sourceforge.net/kb/tcp_slave_howto.html

Começou-se numa 1ª tentativa pelo *Master* e enviar mensagens para um servidor TCP no programa Hercules, com IP e porto parametrizados. A comunicação realizou-se com sucesso.

Exportou-se o projeto de *Slave* para “jar” executável, para mais tarde colocar no iMod e poder executá-lo.


2.3 Execução e análise do projeto final

Colocou-se o ficheiro “jar” do projeto *Slave* no iMod utilizando o Notepad++, que permite acesso FTP ao iMod e criar pastas e adicionar ficheiros.

Neste caso, criou-se uma pasta com “modbusTCP” no caminho /mnt/mtd e depois adicionou-se o ficheiro “jar” nessa pasta (ServidorModbusBRA.jar).

Abriu-se a linha de comandos do Windows e ligou-se ao iMod utilizando o Telnet: comando “telnet” + “IP do iMod”. Neste caso “telnet 192.168.102.53”, seguido com as credenciais de acesso (Login: “user”; Password: “user”); e finalmente foi introduzido o comando “su”, para estar como utilizador “root” e ter mais permissões, em que a *password* é “techbase”.

Introduziu-se o comando “cd /mnt/mtd/modbusTCP/” para se situar na diretoria onde se encontra o executável que pretendemos correr. Ao utilizar o comando “ls”, pode-se listar o conteúdo da diretoria onde se está situado, e neste caso podemos ver que se encontra o ficheiro em questão na diretoria (Figura 25).



```
Telnet 192.168.102.53
techbase login: user
Password:

BusyBox v1.18.4 (2012-06-22 09:17:20 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

[user@techbase ~]# su
Password:
su: incorrect password
[user@techbase ~]# su
Password:

BusyBox v1.18.4 (2012-06-22 09:17:20 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

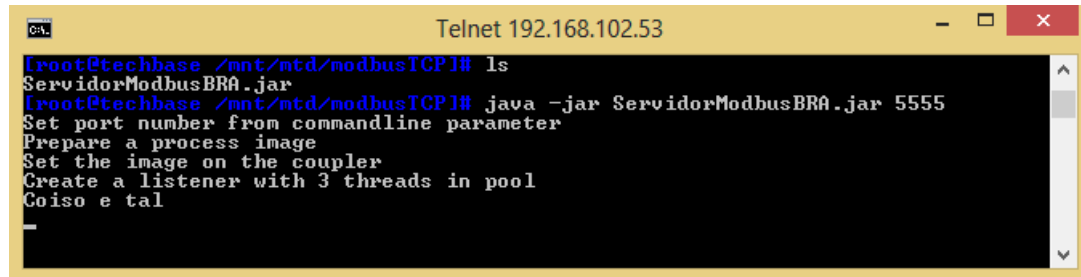
[root@techbase ~]# cd /mnt/mtd/modbusTCP/
[root@techbase /mnt/mtd/modbusTCP]# ls
ServidorModbusBRA.jar
[root@techbase /mnt/mtd/modbusTCP]#
```

Figura 25 – Aceder ao iMod na diretoria onde se encontra o *Slave*

Agora resta apenas correr o executável do *Slave*, que será o servidor para receber pedidos do Master que será corrido no Eclipse.

Para isso, é introduzido o comando “java -jar ServidorModbusBRA.jar 5555”, isto é, “java -jar”, que serve para correr o executável “jar”, seguido com o nome do ficheiro e dos respetivo argumentos da aplicação (neste caso, a porta de valor 5555).

Ao executar é preparado o servidor para estar sob escuta de pedidos provenientes do cliente (Figura 26).



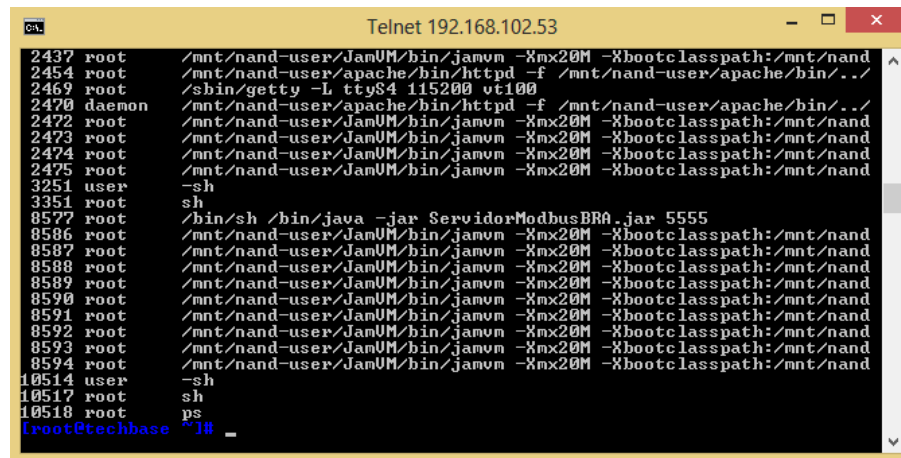
```

Telnet 192.168.102.53
root@techbase /mnt/mtd/modbusTCP1# ls
ServidorModbusBRA.jar
root@techbase /mnt/mtd/modbusTCP1# java -jar ServidorModbusBRA.jar 5555
Set port number from commandline parameter
Prepare a process image
Set the image on the coupler
Create a listener with 3 threads in pool
Coiso e tal

```

Figura 26 – Execução do *Slave*

É possível ver quais os processos que estão a correr no iMod, inclusive para saber se os programas que estamos a executar estão com os processos como pretendemos. Por isso, pode-se usar o comando “ps”, mas como o *prompt* da consola não está disponível, pelo facto de estar preso pelo processo do executável, abre-se outra instância da consola para se verificar os processos correntes (Figura 27).



```

Telnet 192.168.102.53
2437 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
2454 root /mnt/nand-user/apache/bin/httpd -f /mnt/nand-user/apache/bin/..
2469 root /sbin/getty -L ttyS4 115200 vt100
2470 daemon /mnt/nand-user/apache/bin/httpd -f /mnt/nand-user/apache/bin/..
2472 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
2473 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
2474 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
2475 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
3251 user -sh
3351 root sh
8577 root /bin/sh /bin/java -jar ServidorModbusBRA.jar 5555
8586 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
8587 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
8588 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
8589 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
8590 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
8591 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
8592 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
8593 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
8594 root /mnt/nand-user/JamUM/bin/jamvm -Xmx20M -Xbootclasspath:/mnt/nand
10514 user -sh
10517 root sh
10518 root ps
root@techbase ~#

```

Figura 27 - Listagens de processos correntes

No IDE Eclipse, resta agora correr a aplicação *Master* com os respetivos parâmetros/argumentos configurados no projeto, para assim se poder comunicar com o *Slave* que está atualmente em execução.

Para isso deve-se clicar com o botão direito na pasta do projeto *Master* em questão; “*Run As*” > “*Run Configurations*”; tem de se escolher uma instância de configuração de execução do tipo “*Java Application*” (pode-se criar um nova ou editar uma já existente); Na aba “*Arguments*”,

introduz-se os argumentos necessários para a execução da aplicação, que são os seguintes (<> obrigatório, {} opcional):

1. <endereço IP [String]> como InetAddress para a variável “addr”. Opcionalmente o porto pode ser adicionado ao endereço com :<porto>, e lê-lo para a variável “port”.
2. registo [int16] como “int” para a variável “ref”.
3. *bitcount* [int16] como “int” para a variável “count”.
4. {repete [int]} como “int” para a variável “repeat”, com valor 1 por omissão (opcional).

Pode-se então guardar a edição da configuração e executar a aplicação logo de seguida (Figura 28).

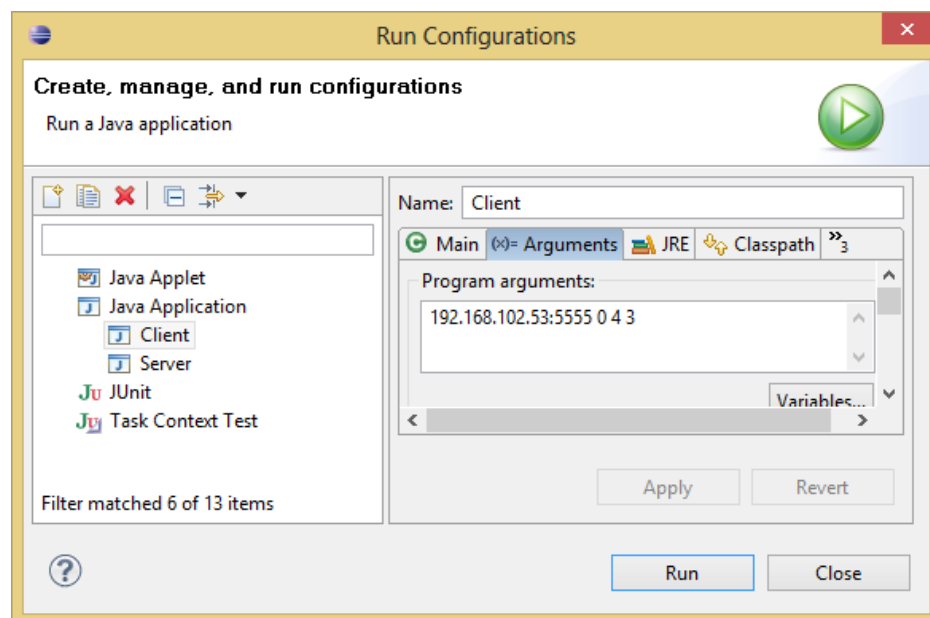


Figura 28 - Configuração da execução da aplicação Master

Notar que o endereço de IP a passar como argumento tem de corresponder ao IP do iMod. Já a porta tem de corresponder àquela passada como argumento na execução do *Slave*.

Na execução são preparados é preparado o módulo de ligação ao servidor e o de transação; depois é preparado o pacote de mensagem para pedido e associado à transação; a transação é efetuada, em que o pacote é enviado ao *Slave*, que envia resposta, e é então recebida a

resposta e imprimida no ecrã (feito N vezes, se passado como o valor for passado como argumento, em que N corresponde a 3 neste caso); e fecha a ligação no final do procedimento.

A execução correu com sucesso e mostrou os valores esperados na consola do Eclipse (Figura 29).

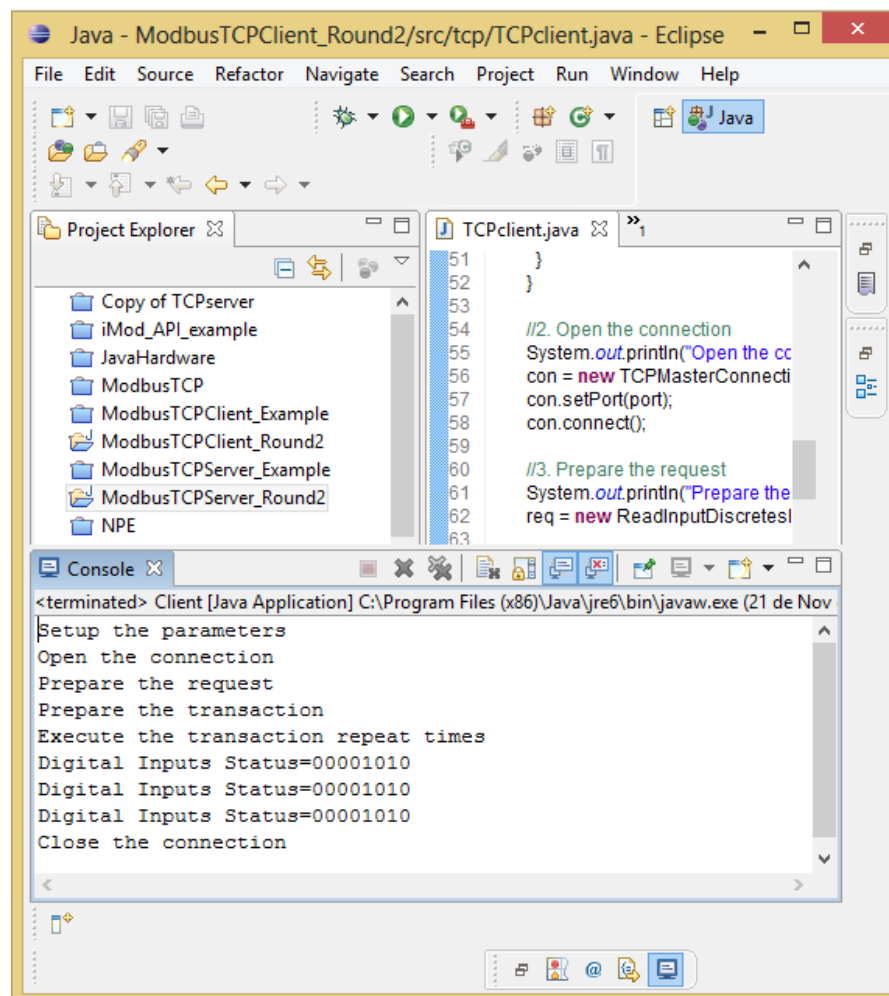


Figura 29 – Execução do Master

O *Master* basicamente leu os valores do tipo *Digital Input* existentes na imagem de processo (*Process Image*) no *Slave*, que por ordem de criação se encontram a partir do bit menos significativo, ou seja, por ordem de criação, estes 4 valores: 0, 1, 0, 1. No final fica como “00001010”.

3 Conclusões

Após a descoberta, exploração e experimentação da biblioteca “jamod”, devido a vantagens já apresentadas, ao bom funcionamento e integração desta no ambiente de desenvolvimento e no iMod, podemos dizer que esta biblioteca é a eleita para utilização no contexto de comunicação utilizando protocolo Modbus TCP.

Além do mais, é possível personalizar alguns módulos para se adaptarem às nossas necessidades, isto é, programar classes que implementem determinado interface e/ou descendam de determinada classes. Pode-se encontrar um exemplo no tópico “*How to make use of the Model*” no seguinte link: <http://jamod.sourceforge.net/kb/processimage.html>

Os projetos encontram-se na pasta do Eclipse comprimida e que está no repositório SVN, em que o respetivo link de caminho é o seguinte: https://svn.isa.pt/svn/iModIntegration/trunk/03Development/Software/Eclipse_com_ModbusTCP.7z

Os últimos projetos *Master* e *Slave* são respetivamente os seguintes: ModbusTCPClient_Round2; ModbusTCPServer_Round2.

C. Anexo C - Especificação de Requisitos Alto Nível



Intelligent Sensing
Anywhere

Especificação

Requisitos Alto Nível

1. Introdução

1.1 Objetivos do documento

O presente documento tem como objetivo apresentar, informação sobre as necessidades do projeto, requisitos no âmbito do projeto de integração de iMod para o Banco.

1.2 Âmbito

Fornecimento e configuração de um sistema centralizado de recolha de dados e dispositivos remotos para recolha e submissão de dados no âmbito da supervisão do consumo de energia na rede bancária do Banco.

1.3 Contexto

O sistema deve permitir a monitorização do consumo de energia do Banco de forma a ser reduzido, evitando desperdícios promovendo uma maior poupança energética e consequentemente económica por parte da empresa.

O objetivo por parte do Banco consiste em reduzir consumos elétricos através de um controlo adequado de todos os sistemas elétricos utilizados nas agências. Será colocado em ação um programa para levar ao cumprimento do objetivo apresentado, em que os principais tópicos do programa são os seguintes:

- Formação aos utilizadores das agências na utilização racional da energia.
- Assegurar a manutenção, comunicações e o funcionamento do sistema de monitorização de consumos que permitirá aferir e quantificar as poupanças.
- Ações de sensibilização e divulgação de boas práticas e comportamentos energéticos mais eficientes, suportado com dados obtidos a partir da monitorização de consumos.

1.4 Ambiente Técnico

Será utilizada uma aplicação de interface Web para o controlo e monitorização dos circuitos de iluminação e AVAC, que representa um servidor central como moderador para recolha e submissão de dados de todas as agências.

Será instalado, no espaço a controlar de cada agência, um conjunto de dispositivos de medição e de atuação que estarão ligados fisicamente via Modbus RTU a um controlador (iMod), que irá conter um protocolo de recolha de informação (iHub). A comunicação será efetuada via GPRS entre cada iMod instalado nas diferentes agências com o servidor central.

Pode-se aceder a uma interface Web do iMod para permitir comunicação entre o iMod e o cliente Web no *browser* de um computador. O iMod e computador ficam ligados por cabo Ethernet a fim de estabelecer a comunicação necessária.

1.5 Acrónimos e Definições

Acrónimo / Definição	Descrição
Agência	Espaço a ser monitorizado (onde se situará o iMod).
Anúncios e Montras	Publicidades.
Ator	A pessoa, organização ou sistema externo ao sistema em desenvolvimento.
AVAC	Aquecimentos, Ventilação e Ar Condicionado.
Caso de Uso	Uma sequência de ações que podem ser traduzidas em valor para o ator.
Chave de Controlo	Permite controlo manual do sistema.
Contadores	Medidor do consumo de energia.
DI/DO	Entradas e saídas digitais.
Diagrama	Uma representação visual do problema ou da solução proposta.
Diagrama de Casos de Uso	Um diagrama que apresenta os casos de uso, atores e as suas interações e relações.
GPRS	Serviço de Rádio de Pacote Geral, tecnologia de transporte de dados por pacotes.
iMod	Equipamento de controlo que irá comunicar através do protocolo iHub.
OneWire	Protocolo de comunicação que reconhece valores dos sensores de temperatura.
Set-point	Valores alterados no sistema central.

Sonda de Temperatura	Dispositivo que por meios mecânicos e elétricos transmite de um ponto a outro a temperatura existente no emissor.
----------------------	---

1.6 Visibilidade do Documento

Este documento é para visualização interna, isto é, apenas os envolvidos neste projeto é que têm acesso a este documento

2. Necessidades

Nome:	NEC01 – Recolha de dados de consumo.	Alterado em:	2014-02-24
Descrição:	O sistema deve recolher dados de consumo das agências através de contadores de eletricidade ligados ao iMod.		

Nome:	NEC02 – Controlar sistemas AVAC.	Alterado em:	2014-02-24
Descrição:	De forma a poder fazer poupança nos consumos, o iMod deve controlar os sistemas AVAC com base nos períodos horários de funcionamento configurados no sistema, bem como nos <i>set-points</i> definidos pelo sistema central.		

Nome:	NEC03 – Controlo da iluminação interior da agência.	Alterado em:	2014-02-24
Descrição:	De forma a poder fazer poupança nos consumos, o iMod deve controlar a iluminação interior com base nos períodos de funcionamento configurados no sistema, bem como através de uma chave de controlo.		

ANEXO C - Especificação de Requisitos Alto Nível

Nome:	NEC04 – Controlo de iluminação dos anúncios/montras.	Alterado em:	2014-02-24
Descrição:	De forma a poder fazer poupança nos consumos, o iMod deve fazer o controlo dos anúncios/montras, será efetuado com base nos períodos de funcionamento configurados no sistema.		

Nome:	NEC05 – Ferramenta de manutenção local.	Alterado em:	2014-02-24
Descrição:	O iMod deve disponibilizar uma interface própria onde o instalador da ISA poderá manipular condições de funcionamento.		

Nome:	NEC06 – Histórico de consumos.	Alterado em:	2014-02-24
Descrição:	Visualização de valores relativos aos consumos dos equipamentos monitorizados.		

Nome:	NEC07 – Emissão de relatórios.	Alterado em:	2014-02-24
Descrição:	Emissão de relatórios de históricos de consumos, análise e processamento de registos e eventos do sistema.		

Nome:	NEC08 – Emissão de configurações e logs.	Alterado em:	2014-02-24
Descrição:	Emissão de configurações e logs da agência para um repositório remoto.		

Nome:	NEC09 – Protocolo de comunicação entre o iMod e o servidor central.	Alterado em:	2014-02-24
Descrição:	Protocolo de comunicação entre o iMod e o servidor central transmitirá valores e os vários períodos de funcionamento dos diversos aparelhos tal como a identificação e reconhecimento dos mesmos.		

Nome:	NEC10 – Comunicação entre os aparelhos de medição e o iMod.	Alterado em:	2014-02-24
Descrição:	A comunicação entre os aparelhos de medição e o iMod será feita via porta RS-485 usando o protocolo Modbus RTU.		

Nome:	NEC11 – Comunicação do iMod com o servidor central.	Alterado em:	2014-02-24
Descrição:	A comunicação do iMod com o servidor central por meio físico.		

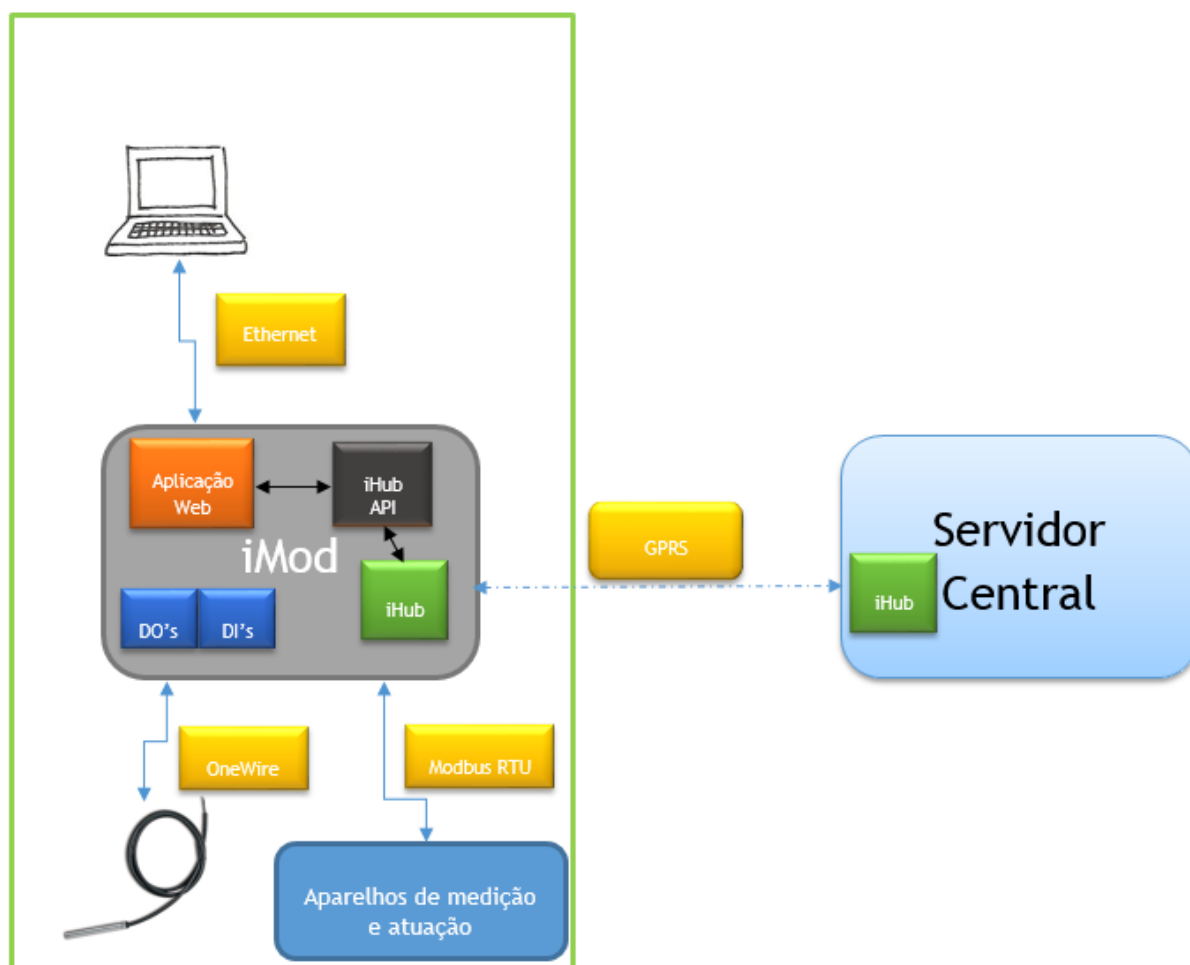
Nome:	NEC12 – Entradas e saídas digitais.	Alterado em:	2014-02-24
Descrição:	Utilizar entradas e saídas digitais do iMod.		

Nome:	NEC13 – Medição de temperatura.	Alterado em:	2014-02-24
Descrição:	Medição de temperatura através de sensores de temperatura.		

Nome:	NEC14 – Permitir a opção de controlo manual sobre os circuitos.	Alterado em:	2014-02-24
Descrição:	Permitir a opção de controlo manual sobre os circuitos, através de uma chave de controlo.		

3. Modelo do sistema

Agência



O iMod inicia a comunicação com o servidor central usando o protocolo iHub via GPRS. Com isto abre um canal de comunicação para transmissão de dados. O iMod lê e escreve parâmetros dos vários aparelhos de medição (sensores, contadores, entre outros), através do protocolo Modbus RTU via porta serial.

Pode-se aceder a uma interface Web do iMod para permitir comunicação entre o iMod e o cliente Web no *browser* de um computador. Para tal será desenvolvida uma API de modo a criar uma ponte de ligação entre a interface Web e o protocolo. O iMod e computador ficam ligados por cabo Ethernet a fim de estabelecer a comunicação necessária.

A comunicação entre o sensor de temperatura e o iMod é conseguido através do protocolo OneWire.

4. Requisitos

Nome:	Req01 – Acesso via login à interface do iMod.	Alterado em:	2014-02-24
Descrição:	<p><i>Login</i> – acesso à interface Web do iMod será através de credenciais de acesso: <i>Username</i> e palavra-passe, iniciando-se no módulo de monitorização.</p> <p><i>Log-out</i>.</p>		

Nome:	Req02 – Módulo <i>monitor</i> .	Alterado em:	2014-02-24
Descrição:	<p>Módulo <i>monitor</i> da interface Web do iMod tem como função permitir ao instalador da ISA uma análise dos dados e acontecimentos da agência no momento.</p>		

Nome:	Req03 – Módulo <i>server configuration</i> .	Alterado em:	2014-02-24
Descrição:	<p>Módulo <i>server configuration</i> do interface Web do iMod, tem como função permitir configurar como é feita a comunicação com um servidor central.</p>		

Nome:	Req04 – Módulo <i>channels configuration</i> .	Alterado em:	2014-02-24
Descrição:	<p>Módulo <i>channels configuration</i> da interface Web do iMod, tem como função permitir configurar os canais que ligaram os dispositivos ao iMod.</p>		

Nome:	Req05 – Módulo <i>data explorer</i> .	Alterado em:	2014-02-24
Descrição:	<p>Módulo <i>data explorer</i> da interface Web do iMod permite ao instalador da ISA recolher informação, analisar históricos de dados, valores e condições de funcionamento da agência.</p>		

Nome:	Req06 – Módulo <i>reports</i> .	Alterado em:	2014-02-24
Descrição:	Módulo <i>reports</i> da interface Web do iMod permite ao instalador da ISA a criação e exportação de relatórios; e a obtenção de <i>templates</i> de documentos para efeitos de manutenção e instalação na agência		

Nome:	Req07 – Módulo <i>alarms</i> .	Alterado em:	2014-02-24
Descrição:	Módulo <i>alarms</i> da interface Web do iMod permite ao instalador da ISA as seguintes operações na agência: -Envio de alarmes (SMS e/ou email) para os técnicos configurados, tendo em atenção o turno em curso; - Parametrização de todos os alarmes; - Priorização dos alarmes (baixo, medio, alto).		

Nome:	Req08 – Exportar configurações.	Alterado em:	2014-02-25
Descrição:	O interface Web do iMod deve permitir ao instalador da ISA poder exportar as configurações desse interface para um repositório remoto.		

Nome:	Req09 – Exportar <i>logs</i> .	Alterado em:	2014-02-25
Descrição:	O interface Web do iMod deve permitir ao instalador da ISA poder exportar os <i>logs</i> desse interface para um repositório remoto.		

Nome:	Req10 – Protocolo de comunicação entre o iMod e o servidor central.	Alterado em:	2014-02-24
Descrição:	Será utilizado para protocolo de comunicação entre o iMod e o servidor, o protocolo iHub que será adaptado programado para ler		

Nome:	Req11 – Comunicação com o iMod via porta serial RS-485.	Alterado em:	2014-02-24
Descrição:	Esta comunicação será estabelecida fisicamente através de porta série RS-485, usando Modbus RTU. Será feito um programa em C para correr em Linux, com a função de receber e enviar informação e terá que ter a possibilidade de configurar o tipo de porta serie, velocidade, bits de paridade, bits de stop.		

Nome:	Req12 – Comunicação iMod com servidor central	Alterado em:	2014-02-24
Descrição:	Será usado o <i>modem</i> GPRS para estabelecer comunicação entre o iMod e o servidor central usando o protocolo iHub.		

Nome:	Req13 – Comunicação entre os equipamentos de medição e o iMod através das entradas digitais do controlador.	Alterado em:	2014-02-24
Descrição:	Comunicação entre os equipamentos de medição e o iMod estabelecido fisicamente às entradas digitais do iMod, recolhendo informação do estado do equipamento.		

Nome:	Req14 – Comunicação entre os equipamentos de medição e o iMod através das saídas digitais do controlador.	Alterado em:	2014-02-24
Descrição:	Comunicação entre os equipamentos de medição e o iMod através das saídas digitais do controlador, ligadas diretamente por meio físico, atuando sobre o equipamento.		

Nome:	Req15 – Comunicação entre sensor de temperatura e o iMod via <i>OneWire</i> .	Alterado em:	2014-02-24
Descrição:	Comunicação entre sensor de temperatura e o iMod via <i>OneWire</i> , consiste numa ligação física entre o sensor e o iMod no qual utiliza o protocolo <i>OneWire</i> que permite reconhecer informação gerada no sensor de temperatura.		

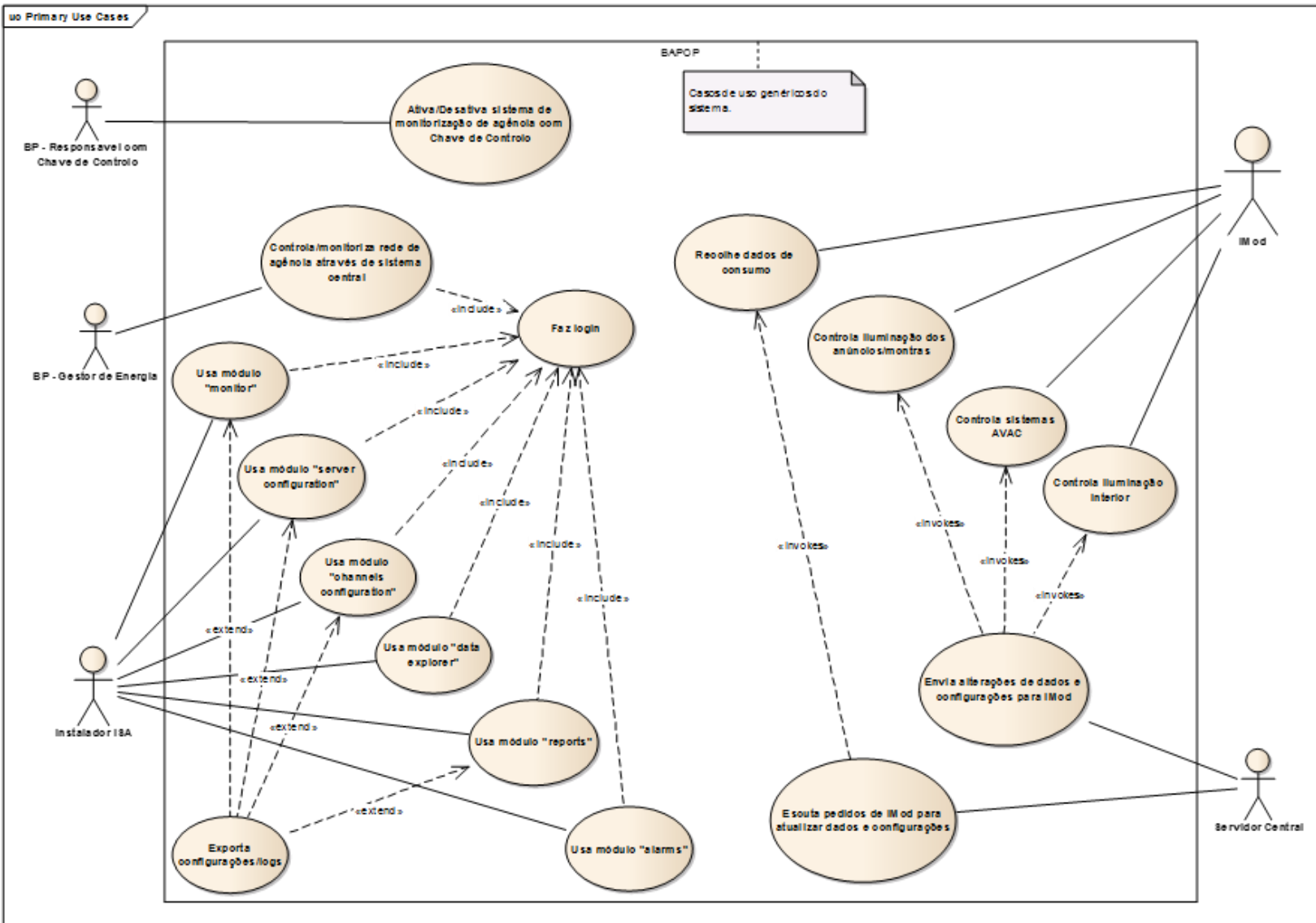
Nome:	Req16 – Chave de controlo.	Alterado em:	2014-02-24
Descrição:	O iMod deverá ter acesso ao estado da chave de controlo (ON/OFF) através de uma entrada digital. Chave de controlo liga e desliga no quadro, todos os circuitos elétricos, permitindo um controlo manual. Quem terá acesso a esta chave será o responsável da agência.		

Nome:	Req17 – Equipamentos ligados ao iMod de cada agência.	Alterado em:	2014-02-24
Descrição:	Poderão ser ligados: - Dois contadores por iMod; - Duas sondas de temperatura por interface OneWire; - Chave de controlo.		

5. Matriz das Necessidades/Requisitos

		Matriz de Rastreabilidade Necessidades/Requisitos do projeto																
		REQ01	REQ02	REQ03	REQ04	REQ05	REQ06	REQ07	REQ08	REQ09	REQ10	REQ11	REQ12	REQ13	REQ14	REQ15	REQ16	REQ17
Recolha de dados de consumo	NEC01																	
Controlar sistemas AVAC	NEC02																	
Controlo da iluminação interior da agência	NEC03																	
Controlo de iluminação dos anúncios/montras	NEC04																	
Ferramenta de manutenção local	NEC05	X	X	X	X	X	X	X	X	X								
Histórico de consumos	NEC06						X											
Emissão de relatórios	NEC07							X										
Emissão de configurações e logs	NEC08								X	X								
Protocolo de comunicação entre o iMod e o servidor central	NEC09										X							
Comunicação entre os aparelhos de medição e o iMod	NEC10											X						
Comunicação do iMod com o servidor central	NEC11												X					
Entradas e saídas digitais	NEC12													X	X			
Medição de temperatura	NEC13															X		X
Permitir a opção de controlo manual sobre os circuitos	NEC14																X	

6. Diagrama de Casos de Uso



D. Anexo D – Mockups para a interface Web do iMod



Figura 30 - Mockup para a página *Login*

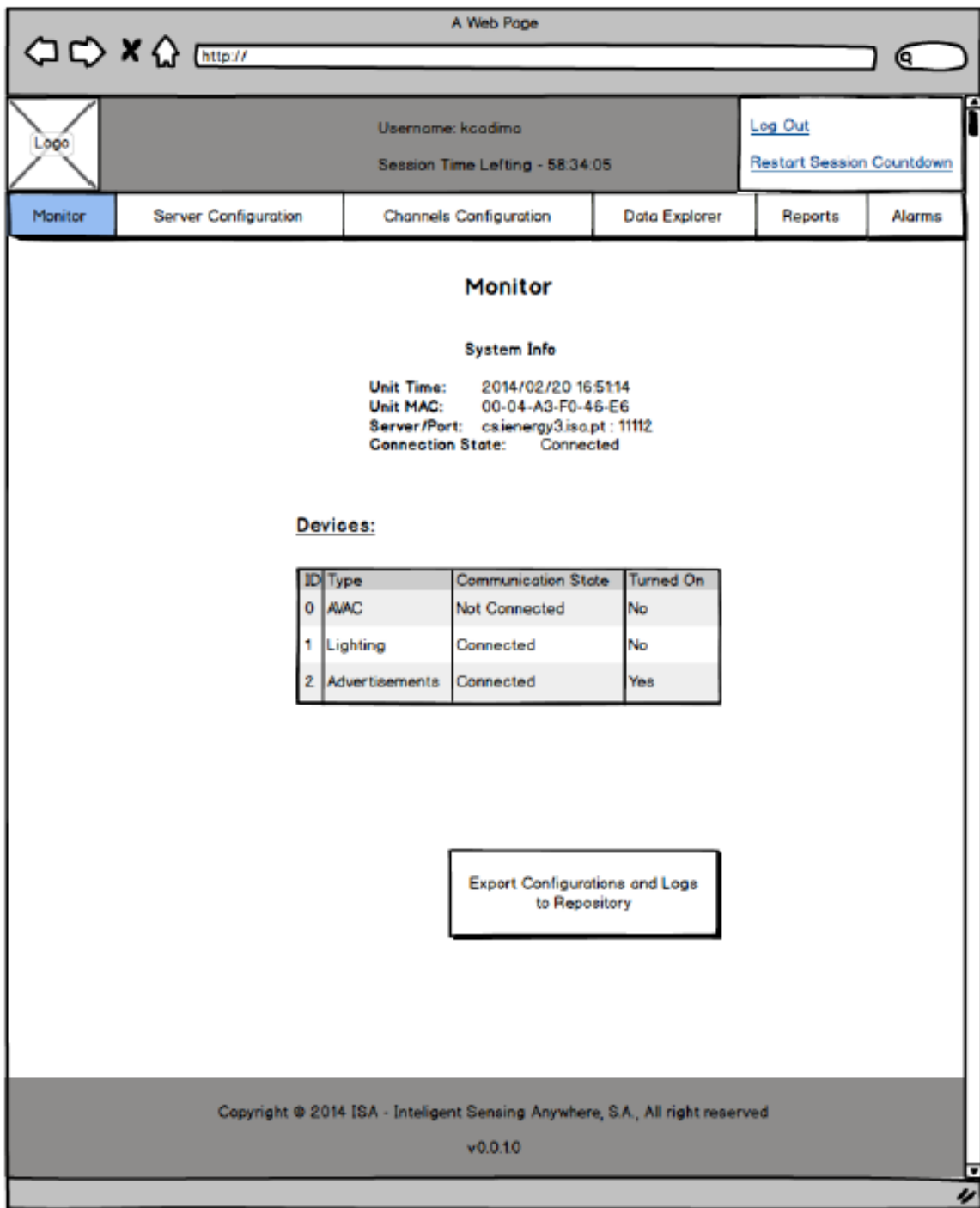


Figura 31 - Mockup para a página Monitor.

Server Configuration

Unit Time:

Unit Configuration: (MAC: 00-04-A3-22-65-A2)	
Serial:	<input type="text" value="162"/>
Server Communication:	
Function as a TCP server:	<input checked="" type="checkbox"/>
(Remote Host:Port)/Local Port:	<input type="text" value="ienegysensorenerbock.pt"/> : <input type="text" value="162"/>
Communication Period:	<input type="text" value="60"/> minutes <input type="button" value="Connect Now"/>
Protocol Framing:	
Encapsulate protocol in frame with CRC:	<input checked="" type="checkbox"/>
Communication Keep Alive Fault Periods: (#Communication errors before stopping the Keep Alive)	<input type="text" value="0"/>
Communication Keep Alive ModBus: (Device ID: Register: Function Code: Value)	<input type="text" value="0"/> : <input type="text" value="0"/> : <input type="text" value="0"/> : <input type="text" value="0"/>
Communication Keep Alive Refresh: (0: Disable Communication Keep Alive)	<input type="text" value="0"/> minutes
Errors before reboot: (0: disable reboot)	<input type="text" value="5"/>

Serial port (RS485) configuration:		#	Port	Baud Rate	Data Bits	Stop Bits	Parity	Resp. Timeout	Query Delay	Retries
Current configurations:	0	tty0	9600	8	1	even	500 ms	415 ms	64	
	1	tty1	9600	8	1	none	1000 ms	50 ms	64	
	2	tty2	9600	8	1	none	1000 ms	50 ms	64	
	3	tty3	9600	8	1	none	1000 ms	50 ms	64	
New configurations:	0	<input type="text" value="tty0"/>	<input type="text" value="9600"/>	8	<input type="text" value="1"/>	<input type="text" value="even"/>	<input type="text" value="500"/> ms	<input type="text" value="415"/> ms	<input type="text" value="64"/>	
	1	<input type="text" value="tty1"/>	<input type="text" value="9600"/>	8	<input type="text" value="1"/>	<input type="text" value="none"/>	<input type="text" value="1000"/> ms	<input type="text" value="50"/> ms	<input type="text" value="64"/>	
	2	<input type="text" value="tty2"/>	<input type="text" value="9600"/>	8	<input type="text" value="1"/>	<input type="text" value="none"/>	<input type="text" value="1000"/> ms	<input type="text" value="50"/> ms	<input type="text" value="64"/>	
	3	<input type="text" value="tty3"/>	<input type="text" value="9600"/>	8	<input type="text" value="1"/>	<input type="text" value="none"/>	<input type="text" value="1000"/> ms	<input type="text" value="50"/> ms	<input type="text" value="64"/>	

Figura 32 - Mockup para a página Server Configuration.

Channels Configuration

Value refresh times (local time): [Cycle start: 2013/10/16 10:55:50; Last value: (remote:)]

Index	Type	ID	Acqu. Period	Additional Data	Last Values
0	None	0	15 min	MBus Conf #: 0 Register: 5002 Format: 32bit FP (IEEE754 single) First Byte/Word: 1 MSB/LSW Scale: 1	-
1	Modbus FC4	10	1 min	MBus Conf #: 1 Register: 5004 Format: 16bit UInt First Byte/Word: 0: MSB/MSW Scale: 1	Time: := ??? =- Value: := ??? =-
2	OneWire	0	15 min	MBus Conf #: 2 Register: 5006 Format: 32bit Int First Byte/Word: 2: LSB/MSW Scale: 1	Time: := ??? =- Value: := ??? =-
3	None	0	15 min	MBus Conf #: 3 Register: 5000 Format: 32 digital inputs First Byte/Word: 3: LSB/LSW Scale: 0.001	-
4	Modbus FC1/5	0	1 min	MBus Conf #: 2 Register: 5006 Format: 32bit Int First Byte/Word: 2: LSB/MSW Scale: 1	Time: := ??? =- Value: := ??? =-
5	Digital Input	0	15 min	MBus Conf #: 3 Register: 5000 Format: 32 digital inputs First Byte/Word: 3: LSB/LSW Scale: 0.001	Time: := ??? =- Value: := ??? =-
6	None	0	15 min	MBus Conf #: 3 Register: 5000 Format: 32 digital inputs First Byte/Word: 3: LSB/LSW Scale: 0.001	-
7	Modbus FC2	0	1 min	MBus Conf #: 3 Register: 5000 Format: 32 digital inputs First Byte/Word: 3: LSB/LSW Scale: 0.001	Time: := ??? =- Value: := ??? =-
8	Digital Output	0	15 min	MBus Conf #: 3 Register: 5000 Format: 32 digital inputs First Byte/Word: 3: LSB/LSW Scale: 0.001	Time: := ??? =- Value: := ??? =-
9	None	0	15 min	MBus Conf #: 3 Register: 5000 Format: 32 digital inputs First Byte/Word: 3: LSB/LSW Scale: 0.001	-
10	Modbus FC3/6	0	1 min	MBus Conf #: 3 Register: 5000 Format: 32 digital inputs First Byte/Word: 3: LSB/LSW Scale: 0.001	Time: := ??? =- Value: := ??? =-
11	None	0	15 min	MBus Conf #: 3 Register: 5000 Format: 32 digital inputs First Byte/Word: 3: LSB/LSW Scale: 0.001	-

Figura 33 - Mockup para a página Channels Configuration. As cruzes vermelhas significam que ainda estavam para ser definidos os atributos em questão para determinados tipos de parâmetros de canais.

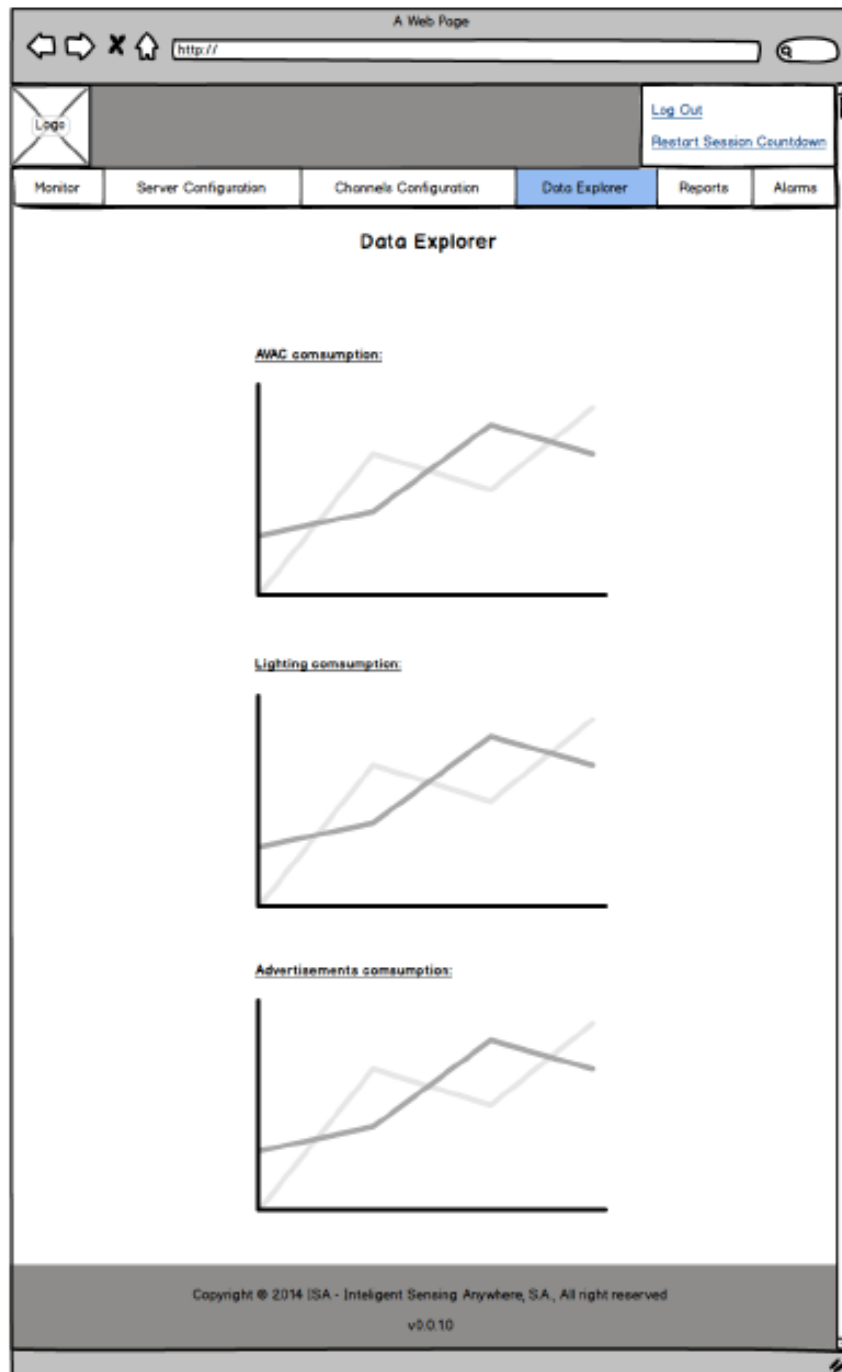


Figura 34 - Mockup da página *Data Explorer*

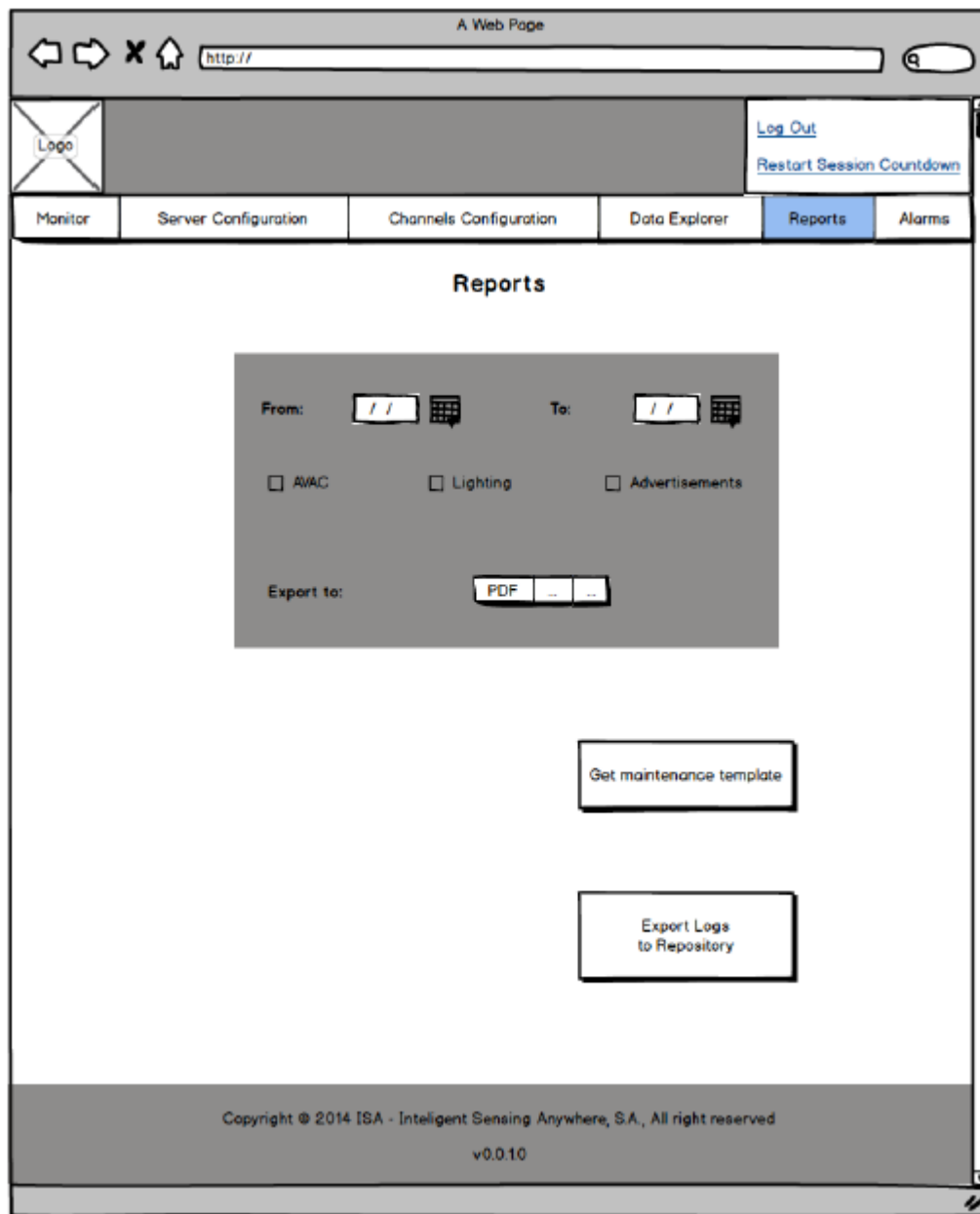


Figura 35 - Mockup da página Reports

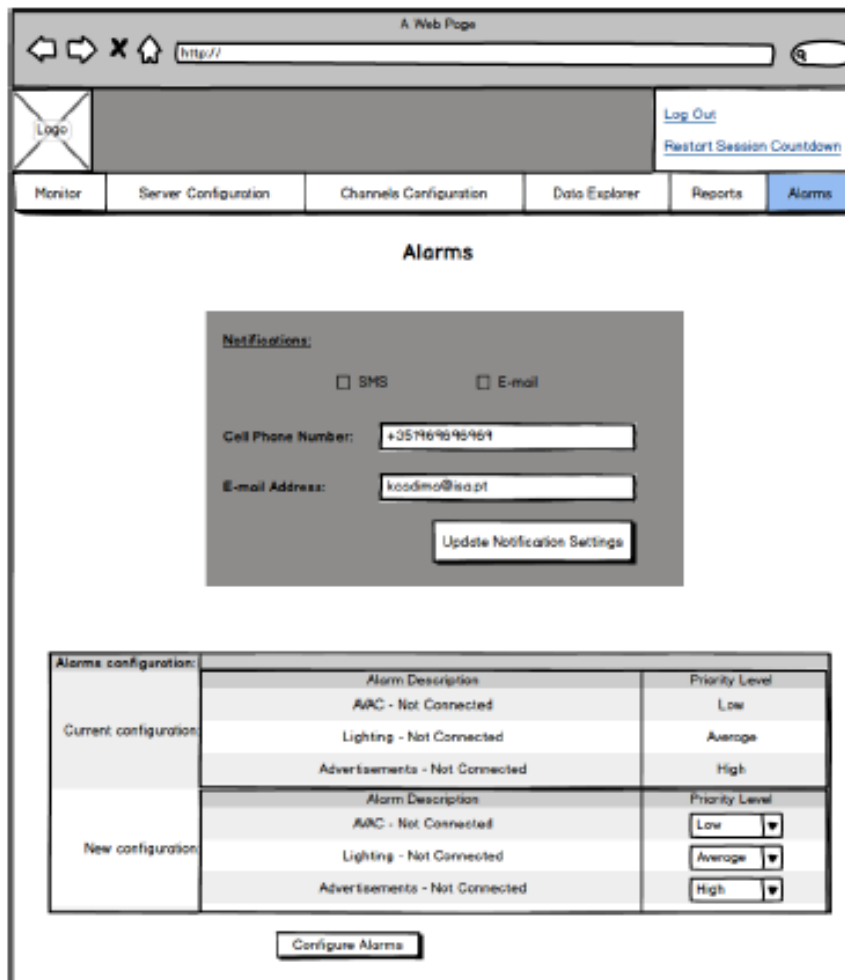


Figura 36 - Mockup da página Alarms

E. Anexo E - Plano de Testes da interface Web do iMod

1. Sessão de utilizador e navegação entre páginas

#	Type	Description
C1	Scenario	1. (S) Access the Website (per default the session is not logged in) - the Login page is presented.
1.1	Test	2. (S) Click on the "Log in" button. 3. (T) Verify that the message "Username and/or password lefting!" appears on the screen.
1.2	Test	2. (S) Insert the text "test" (random text) in the field Username. 3. (S) Click on the "Log in" button. 4. (T) Verify that the message "Username and/or password lefting!" appears on the screen.
1.3	Test	2. (S) Insert the text "test" (random text) in the field Password. 3. (S) Click on the "Log in" button. 4. (T) Verify that the message "Username and/or password lefting!" appears on the screen.
1.4	Test	2. (S) Insert the text "kcadima" (username must be set in the software's configuration file) in the field Username. 3. (S) Click on the "Log in" button. 4. (T) Verify that the message "Username and/or password lefting!" appears on the screen.
1.5	Test	2. (S) Insert the text "1234" (password must be set in the software's configuration file with MD5 encryption) in the field Password. 3. (S) Click on the "Log in" button. 4. (T) Verify that the message "Username and/or password lefting!" appears on the screen.
1.6	Test	2. (S) Insert the text "kcadima" (username must be set in the software's configuration file) in the field Username. 3. (S) Insert the text "test" (random text) in the field Password. 4. (S) Click on the "Log in" button. 5. (T) Verify that the message "Wrong username and/or password!" appears on the screen.
1.7	Test	2. (S) Insert the text "test" (random text) in the field Username. 3. (S) Insert the text "1234" (password must be set in the software's configuration file with MD5 encryption) in the field Password. 4. (S) Click on the "Log in" button. 5. (T) Verify that the message "Wrong username and/or password!" appears on the screen.

1.8	Test	<ol style="list-style-type: none"> 2. (S) Insert the text "test" (random text) in the field Username. 3. (S) Insert the text "test2" (random text) in the field Password. 4. (S) Click on the "Log in" button. 5. (T) Verify that the message "Wrong username and/or password!" appears on the screen.
1.9	Test	<ol style="list-style-type: none"> 2. (S) Insert the text "kcadima" (username must be set in the software's configuration file) in the field Username. 3. (S) Insert the text "1234" (password must be set in the software's configuration file with MD5 encryption) in the field Password. 4. (S) Click on the "Log in" button. 5. (T) Verify that the Monitor page is shown.
1.10	Test	<ol style="list-style-type: none"> 2. (S) Insert in the browser's URL field the text "monitor.php" after the project's name (Ex: localhost/imodinterface/monitor.php) and submit it. 3. (T) Verify that the Login page is shown (page "login.php"). <p>PS: Repeat the test with step 2 using the following pages instead:</p> <ul style="list-style-type: none"> - serverconf.php - channelsconf.php - dataexplorer.php - reports.php - alarms.php - generalconf.php
C2	Scenario	<ol style="list-style-type: none"> 1. (S) Log in (follow steps from Test 1.9)
2.1	Test	<ol style="list-style-type: none"> 2. (S) Click on the "Monitor" button to go to the page "monitor.php" (by default this page must be shown). 3. (S) Insert in the browser's URL field the text "login.php" after the project's name (Ex: localhost/imodinterface/login.php) and submit it. 4. (T) Verify that the Monitor page is shown (page "monitor.php"). <p>PS: Repeat the test with step 2 using the following pages instead:</p> <ul style="list-style-type: none"> - "Server Configuration" button to go to the page "serverconf.php" - "Channels Configuration" button to go to the page "channelsconf.php" - "Data Explorer" button to go to the page "dataexplorer.php" - "Reports" button to go to the page "reports.php" - "Alarms" button to go to the page "alarms.php" - "General Configuration" button to go to the page "generalconf.php"
2.2	Test	<ol style="list-style-type: none"> 2. (S) Click on "Log out" (on the top-left corner of the screen). 3. (T) Verify that the Login page is shown (page "login.php").
C3	Scenario	<ol style="list-style-type: none"> 1. (S) Set the value of the parameter <code>\$confs['sessiontime']</code> to 30, in the file <code>conf/conf.php</code> and save the changes (the session time is set to 30 seconds to be more comfortable while testing).

		2. (S) Log in (follow steps from Test 1.9)
3.1	Test	3. (S) Wait until the session countdown in the header reaches the 0 (zero) seconds. 4. (T) Verify that the Login page is shown (page "login.php").
3.2	Test	3. (S) Wait 10 seconds. 4. (S) Click on "Restart Session Countdown" (on the top-left corner of the screen). 5. (T) Verify that the session countdown in the header is reset to 30 seconds.
3.3	Test	3. (T) Verify that the session countdown starts with "30 seconds".
3.4	Test	3. (S) Close the browser. 4. (S) Wait 30 seconds. 5. (S) Re-open the browser, write the project URL and submit it. 6. (T) Verify that the Login page is shown (page "login.php").
3.5	Test	3. (T) Notice that the session countdown is counting down.

2. Página de Monitorização

Testes não definidos para esta página.

3. Página de Configuração de Servidor

#	Type	Description
C1	Scenario	1. (S) Log in (see Scenario C2 in the Session sheet) 2. (S) Click on the "Server Configuration" button to go to the page "serverconf.php".
C2	Scenario	1. (S) Scenario C1 2. (S) "Unit Time" form
2.1	Test	3. (S) Click on the "Configure Unit" button. 4. (T) Verify that the message "SUCCESS: Unit time configured with success." appears on the screen. The "Unit time" date value must be the same than the "Local time" one when the button was clicked
C3	Scenario	1. (S) Scenario C1 2. (S) "Test Connection" form 3. (S) "(Remote Host:Port)" field

3.1	Test	<p>4. (S) Insert the text "123" in the "(Remote Host:Port)" field.</p> <p>5. (S) Click on the "Connect Now" button.</p> <p>6. (T) Verify that the message "Please match the requested format." appears on the screen.</p>
3.2	Test	<p>4. (S) Insert the text "123.123.123" in the "(Remote Host:Port)" field.</p> <p>5. (S) Click on the "Connect Now" button.</p> <p>6. (T) Verify that the message "Please match the requested format." appears on the screen.</p>
3.3	Test	<p>4. (S) Insert the text "isa" in the "(Remote Host:Port)" field.</p> <p>5. (S) Click on the "Connect Now" button.</p> <p>6. (T) Verify that the message "Please match the requested format." appears on the screen.</p>
3.4	Test	<p>4. (S) Let the field "(Remote Host:Port)" empty.</p> <p>5. (S) Click on the "Connect Now" button.</p> <p>6. (T) Verify that the message "Please fill out this field." appears on the screen.</p>
3.5	Test	<p>4. (S) Insert the text "123.123.123.123" in the field "(Remote Host:Port)" (assuming this IP address is inexistent/unavailable).</p> <p>5. (S) Click on the "Connect Now" button.</p> <p>6. (T) Verify that the message "WARNING: Test connection - connection not made.</p> <p>Failures: Connection not made - host: 123.123.123.123 - port: (SET PORT)" appears on the screen. The text "123.123.123.123" still appears in the previously set field due to unsuccessful operation.</p>
3.6	Test	<p>4. (S) Insert the text "teste.isa.pt" in the field "(Remote Host:Port)" (assuming this DNS address is inexistent/unavailable).</p> <p>5. (S) Click on the "Connect Now" button.</p> <p>6. (T) Verify that the message "WARNING: Test connection - connection not made.</p> <p>Failures: Connection not made - host: teste.isa.pt - port: (SET PORT)" appears on the screen. The text "teste.isa.pt" still appears in the previously set field due to unsuccessful operation.</p>
C4	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Test Connection" form</p> <p>3. (S) "(Local Port)" field</p>
4.1	Test	<p>4. (S) Insert the number -1 in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be greater than or equal to 1." appears on the screen.</p>
4.2	Test	<p>4. (S) Insert the number 0 in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be greater than or equal to 1." appears on the screen.</p>

4.3	Test	<p>4. (S) Insert the number 65536 in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be less than or equal to 65535." appears on the screen.</p>
4.4	Test	<p>4. (S) Insert the text "1a" in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please enter a number." appears on the screen.</p>
4.5	Test	<p>4. (S) Let the field "(Local Port)" empty.</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please fill out this field." appears on the screen.</p>
4.6	Test	<p>4. (S) Insert the number 33 in the field "(Local Port)" (assuming the connection will be inexistent/unavailable).</p> <p>5. (S) Click on the "Connect Now" button.</p> <p>6. (T) Verify that the message "WARNING: Test connection - connection not made. Failures: Connection not made - host: (SET HOST) - port: 33" appears on the screen. The value 33 still appears in the previously set field due to unsuccessful operation.</p>
C5	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Unit Configuration" form</p> <p>3. (S) "Serial" field</p>
5.1	Test	<p>4. (S) Insert the number -1 in the field "Serial".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.</p>
5.2	Test	<p>4. (S) Insert the number 256 in the field "Serial".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be less than or equal to 255." appears on the screen.</p>
5.3	Test	<p>4. (S) Insert the text "1a" in the field "Serial".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please enter a number." appears on the screen.</p>
5.4	Test	<p>4. (S) Let the field "Serial" empty.</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please fill out this field." appears on the screen.</p>
5.5	Test	<p>4. (S) Insert an integer number different from the existent one in the field "Serial" empty (value from 0 to 255).</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The previously set field must show the new written value.</p>

C6	Scenario	<ol style="list-style-type: none"> 1. (S) Scenario C1 2. (S) "Unit Configuration" form 3. (S) "Function as a TCP server" checkbox
6.1	Test	<ol style="list-style-type: none"> 4. (S) "Function as a TCP server" checkbox is unchecked and "(Remote Host:Port)" field is enabled. 5. (S) Click on the "Function as a TCP server" checkbox to check it. 6. (T) Verify that the "(Remote Host:Port)" field is disabled. <p>PS: If at the step 4 "Function as a TCP server" checkbox is checked and "(Remote Host:Port)" field is disabled, try test 6.2 first.</p>
6.2	Test	<ol style="list-style-type: none"> 4. (S) "Function as a TCP server" checkbox is checked and "(Remote Host:Port)" field is disabled. 5. (S) Click on the "Function as a TCP server" checkbox to uncheck it. 6. (T) Verify that the "(Remote Host:Port)" field is enabled. <p>PS: If at the step 4 "Function as a TCP server" checkbox is unchecked and "(Remote Host:Port)" field is enabled, try test 6.1 first.</p>
6.3	Test	<ol style="list-style-type: none"> 4. (S) Check "Function as a TCP server" checkbox. 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The previously checked checkbox must appear checked. Verify that the "(Remote Host:Port)" field is disabled.
6.4	Test	<ol style="list-style-type: none"> 4. (S) Uncheck "Function as a TCP server" checkbox. 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The previously unchecked checkbox must appear unchecked. Verify that the "(Remote Host:Port)" field is enabled.
C7	Scenario	<ol style="list-style-type: none"> 1. (S) Scenario C1 2. (S) "Unit Configuration" form 3. (S) "(Remote Host:Port)" field is enabled
7.1	Test	<ol style="list-style-type: none"> 4. (S) Insert the text "123" in the "(Remote Host:Port)" field. 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Please match the requested format." appears on the screen.
7.2	Test	<ol style="list-style-type: none"> 4. (S) Insert the text "123.123.123" in the "(Remote Host:Port)" field. 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Please match the requested format." appears on the screen.
7.3	Test	<ol style="list-style-type: none"> 4. (S) Insert the text "isa" in the "(Remote Host:Port)" field. 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Please match the requested format." appears on the screen.

7.4	Test	<p>4. (S) Let the field "(Remote Host:Port)" empty.</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please fill out this field." appears on the screen.</p>
7.5	Test	<p>4. (S) Insert the text "123.123.123.123" in the field "(Remote Host:Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The text "123.123.123.123" must appear in the previously set field.</p>
7.6	Test	<p>4. (S) Insert the text "teste.isa.pt" in the field "(Remote Host:Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The text "teste.isa.pt" must appear in the previously set field.</p>
C8	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Unit Configuration" form</p> <p>3. (S) "(Local Port)" field</p>
8.1	Test	<p>4. (S) Insert the number -1 in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be greater than or equal to 1." appears on the screen.</p>
8.2	Test	<p>4. (S) Insert the number 0 in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be greater than or equal to 1." appears on the screen.</p>
8.3	Test	<p>4. (S) Insert the number 65536 in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be less than or equal to 65535." appears on the screen.</p>
8.4	Test	<p>4. (S) Insert the text "1a" in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please enter a number." appears on the screen.</p>
8.5	Test	<p>4. (S) Let the field "(Local Port)" empty.</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please fill out this field." appears on the screen.</p>
8.6	Test	<p>4. (S) Insert the number 33 in the field "(Local Port)".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The value 33 must appear in the previously set field.</p>
C9	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Unit Configuration" form</p>

		3. (S) "Communication Period" field
9.1	Test	4. (S) Insert the number -1 in the field "Communication Period". 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Value must be greater than or equal to 1." appears on the screen.
9.2	Test	4. (S) Insert the number 0 in the field "Communication Period". 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Value must be greater than or equal to 1." appears on the screen.
9.3	Test	4. (S) Insert the number 65521 in the field "Communication Period". 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Value must be less than or equal to 65520." appears on the screen.
9.4	Test	4. (S) Insert the text "1a" in the field "Communication Period". 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Please enter a number." appears on the screen.
9.5	Test	4. (S) Let the field "Communication Period" empty. 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Please fill out this field." appears on the screen.
9.6	Test	4. (S) Insert the number 59 in the field "Communication Period". 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Please enter a number multiple or submultiple of 60." appears on the screen and the field "Communication Period" is focused.
9.7	Test	4. (S) Insert the number 121 in the field "Communication Period". 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "Please enter a number multiple or submultiple of 60." appears on the screen and the field "Communication Period" is focused.
9.8	Test	4. (S) Insert the number 30 in the field "Communication Period". 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The value 30 must appear in the previously set field.
C10	Scenario	1. (S) Scenario C1 2. (S) "Unit Configuration" form 3. (S) "Encapsulate protocol in frame with CRC" checkbox
10.1	Test	4. (S) Check "Encapsulate protocol in frame with CRC" checkbox. 5. (S) Click on the "Configure Unit" button. 6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on

		the screen. The previously checked checkbox must appear checked.
10.2	Test	<p>4. (S) Uncheck "Encapsulate protocol in frame with CRC" checkbox.</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The previously unchecked checkbox must appear unchecked.</p>
C11	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Unit Configuration" form</p> <p>3. (S) "Errors before reboot" field</p>
11.1	Test	<p>4. (S) Insert the number -1 in the field "Errors before reboot".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.</p>
11.2	Test	<p>4. (S) Insert the number 256 in the field "Errors before reboot".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Value must be less than or equal to 255." appears on the screen.</p>
11.3	Test	<p>4. (S) Insert the text "1a" in the field "Errors before reboot".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please enter a number." appears on the screen.</p>
11.4	Test	<p>4. (S) Let the field "Errors before reboot" empty.</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "Please fill out this field." appears on the screen.</p>
11.5	Test	<p>4. (S) Insert the number 33 in the field "Communication Period".</p> <p>5. (S) Click on the "Configure Unit" button.</p> <p>6. (T) Verify that the message "SUCCESS: Unit configuration configured with success." appears on the screen. The value 33 must appear in the previously set field.</p>
C12	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Serial Port Configuration" form</p> <p>3. (S) "New configurations" row</p> <p>4. (S) "Port" attribute column</p>
12.1	Test	<p>5. (S) At the row #0, select a different value from the already selected one, in the "Port" selection field.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "SUCCESS: Serial ports configured with success." appears on the screen and that the new selected value appears in its proper place.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>

C13	Scenario	<ol style="list-style-type: none"> 1. (S) Scenario C1 2. (S) "Serial Port Configuration" form 3. (S) "New configurations" row 4. (S) "Baud Rate" attribute column
13.1	Test	<ol style="list-style-type: none"> 5. (S) Insert the number 2399 in the field "Baud Rate" custom field. 6. (S) Click on the "Configure RS485" button. 7. (T) Verify that the message "Value must be greater than or equal to 2400." appears on the screen. <p>PS: - Repeat the test for rows #1, #2 and #3. - If at the step 5 "Baud Rate" selection field has a number value selected and the custom field below is disabled, try test 13.5 first.</p>
13.2	Test	<ol style="list-style-type: none"> 5. (S) Insert the number 115201 in the field "Baud Rate" custom field. 6. (S) Click on the "Configure RS485" button. 7. (T) Verify that the message "Value must be less than or equal to 115200." appears on the screen. <p>PS: - Repeat the test for rows #1, #2 and #3. - If at the step 5 "Baud Rate" selection field has a number value selected and the custom field below is disabled, try test 13.5 first.</p>
13.3	Test	<ol style="list-style-type: none"> 4. (S) Insert the text "1a" in the field "Baud Rate" custom field. 5. (S) Click on the "Configure RS485" button. 6. (T) Verify that the message "Please enter a number." appears on the screen. <p>PS: - Repeat the test for rows #1, #2 and #3. - If at the step 5 "Baud Rate" selection field has a number value selected and the custom field below is disabled, try test 13.5 first.</p>
13.4	Test	<ol style="list-style-type: none"> 4. (S) Let the field "Baud Rate" custom field empty. 5. (S) Click on the "Configure RS485" button. 6. (T) Verify that the message "Please fill out this field." appears on the screen. <p>PS: - Repeat the test for rows #1, #2 and #3. - If at the step 5 "Baud Rate" selection field has a number value selected and the custom field below is disabled, try test 13.5 first.</p>
13.5	Test	<ol style="list-style-type: none"> 5. (S) At the row #0, the selected value of the selection field of the "Baud Rate" is a number and the custom field below has the same number and is disabled. 6. (S) Select "Custom" in the "Baud Rate" selection field of the row #0. 7. (T) Verify that the custom field below the field from step 6 is enabled. <p>PS: - Repeat the test for rows #1, #2 and #3. - If at the step 5 "Baud Rate" selection field has the value "Custom" selected and the custom field</p>

		below is enabled, try test 13.6 first.
13.6	Test	<p>5. (S) At the row #0, the selected value of the selection field of the "Baud Rate" is "Custom" and the custom field below has a number and is enabled.</p> <p>6. (S) Select a number in the "Baud Rate" selection field of the row #0.</p> <p>7. (T) Verify that the custom field below the field from step 6 is disabled.</p> <p>PS: - Repeat the test for rows #1, #2 and #3. - If at the step 5 "Baud Rate" selection field has a number value selected and the custom field below is disabled, try test 13.5 first.</p>
13.7	Test	<p>5. (S) At the row #0, select a different value from the already selected one, in the "Baud Rate" selection field.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "SUCCESS: Serial ports configured with success." appears on the screen and that the new selected value appears in its proper place.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
13.8	Test	<p>5. (S) At the row #0, insert a different integer value from the already existent one, in the "Baud Rate" custom field.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "SUCCESS: Serial ports configured with success." appears on the screen and that the new inserted value appears in its proper place.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
C14	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Serial Port Configuration" form</p> <p>3. (S) "New configurations" row</p> <p>4. (S) "Stop Bits" attribute column</p>
14.1	Test	<p>5. (S) At the row #0, select a different value from the already selected one, in the "Stop Bits" selection field.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "SUCCESS: Serial ports configured with success." appears on the screen and that the new selected value appears in its proper place.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
C15	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Serial Port Configuration" form</p> <p>3. (S) "New configurations" row</p> <p>4. (S) "Parity" attribute column</p>

15.1	Test	<p>5. (S) At the row #0, select a different value from the already selected one, in the "Parity" selection field.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "SUCCESS: Serial ports configured with success." appears on the screen and that the new selected value appears in its proper place.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
C16	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) "Serial Port Configuration" form</p> <p>3. (S) "New configurations" row</p> <p>4. (S) "Resp. Timeout" attribute column</p>
16.1	Test	<p>5. (S) At the row #0, insert the number -1 in the field "Resp. Timeout".</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
16.2	Test	<p>5. (S) At the row #0, insert the number 65536 in the field "Resp. Timeout".</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "Value must be less than or equal to 65535." appears on the screen.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
16.3	Test	<p>5. (S) At the row #0, insert the text "1a" in the field "Resp. Timeout".</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "Please enter a number." appears on the screen.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
16.4	Test	<p>5. (S) At the row #0, let the field "Resp. Timeout" empty.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "Please fill out this field." appears on the screen.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
16.5	Test	<p>5. (S) At the row #0, insert a different integer value from the already existent one, in the "Resp. Timeout" field.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "SUCCESS: Serial ports configured with success." appears on the screen and that the new inserted value appears in its proper place.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>

C17	Scenario	<ol style="list-style-type: none"> 1. (S) Scenario C1 2. (S) "Serial Port Configuration" form 3. (S) "New configurations" row 4. (S) "Query Delay" attribute column
17.1	Test	<ol style="list-style-type: none"> 5. (S) At the row #0, insert the number -1 in the field "Query Delay". 6. (S) Click on the "Configure RS485" button. 7. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen. <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
17.2	Test	<ol style="list-style-type: none"> 5. (S) At the row #0, insert the number 65536 in the field "Query Delay". 6. (S) Click on the "Configure RS485" button. 7. (T) Verify that the message "Value must be less than or equal to 65535." appears on the screen. <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
17.3	Test	<ol style="list-style-type: none"> 5. (S) At the row #0, insert the text "1a" in the field "Query Delay". 6. (S) Click on the "Configure RS485" button. 7. (T) Verify that the message "Please enter a number." appears on the screen. <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
17.4	Test	<ol style="list-style-type: none"> 5. (S) At the row #0, let the field "Query Delay" empty. 6. (S) Click on the "Configure RS485" button. 7. (T) Verify that the message "Please fill out this field." appears on the screen. <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
17.5	Test	<ol style="list-style-type: none"> 5. (S) At the row #0, insert a different integer value from the already existent one, in the "Query Delay" field. 6. (S) Click on the "Configure RS485" button. 7. (T) Verify that the message "SUCCESS: Serial ports configured with success." appears on the screen and that the new inserted value appears in its proper place. <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
C18	Scenario	<ol style="list-style-type: none"> 1. (S) Scenario C1 2. (S) "Serial Port Configuration" form 3. (S) "New configurations" row 4. (S) "Retries" attribute column

18.1	Test	<p>5. (S) At the row #0, insert the number -1 in the field "Retries".</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
18.2	Test	<p>5. (S) At the row #0, insert the number 100 in the field "Retries".</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "Value must be less than or equal to 99." appears on the screen.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
18.3	Test	<p>5. (S) At the row #0, insert the text "1a" in the field "Retries".</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "Please enter a number." appears on the screen.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
18.4	Test	<p>5. (S) At the row #0, let the field "Retries" empty.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "Please fill out this field." appears on the screen.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
18.5	Test	<p>5. (S) At the row #0, insert a different integer value from the already existent one, in the "Retries" field.</p> <p>6. (S) Click on the "Configure RS485" button.</p> <p>7. (T) Verify that the message "SUCCESS: Serial ports configured with success." appears on the screen and that the new inserted value appears in its proper place.</p> <p>PS: - Repeat the test for rows #1, #2 and #3.</p>
C19	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) Export Server Configuration to Repository</p>
19.1	Test	

4. Página de Configuração de Canais

#	Type	Description
C1	Scenario	1. (S) Log in (see Scenario C2 in the Session sheet)

		2. (S) Click on the "Channels Configuration" button to go to the page "channelsconf.php".
C2	Scenario	1. (S) Scenario C1 2. (S) Channels configuration table with at least 3 rows. 3. (S) In the row with index number 0 (repeat this test later with row 1 and the last row). 4. (S) "ID" field
2.1	Test	5. (S) Insert the number -1 in the field "ID". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be greater than or equal to 1." appears on the screen.
2.2	Test	5. (S) Insert the number 0 in the field "ID". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be greater than or equal to 1." appears on the screen.
2.3	Test	5. (S) Insert the number 255 in the field "ID". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be less than or equal to 254." appears on the screen.
2.4	Test	5. (S) Insert the text "1a" in the field "ID". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please enter a number." appears on the screen.
2.5	Test	5. (S) Let the field "ID" empty. 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please fill out this field." appears on the screen.
C3	Scenario	1. (S) Scenario C1 2. (S) Channels configuration table with at least 3 rows. 3. (S) In the row with index number 0 (repeat this test later with row 1 and the last row). 4. (S) "Acqu. Period" field
3.1	Test	5. (S) Insert the number -1 in the field "Acqu. Period". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.
3.2	Test	5. (S) Insert the number 241 in the field "Acqu. Period". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be less than or equal to 240." appears on the screen.
3.3	Test	5. (S) Insert the text "1a" in the field "Acqu. Period". 6. (S) Click on the "Configure All" button.

		7. (T) Verify that the message "Please enter a number." appears on the screen.
3.4	Test	5. (S) Let the field "Acqu. Period" empty. 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please fill out this field." appears on the screen.
C4	Scenario	1. (S) Scenario C1 2. (S) Channels configuration table with at least 3 rows. 3. (S) In the row with index number 0 (repeat this test later with row 1 and the last row). 4. (S) "Register" field ("Additional Data" column), in the case of being a parameter of a Modbus type or OneWire type (case not, choose one of the Modbus types or OneWire in the selection field).
4.1	Test	5. (S) Insert the number -1 in the field "Register". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.
4.2	Test	5. (S) Insert the number 65536 in the field "Register". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be less than or equal to 65535." appears on the screen.
4.3	Test	5. (S) Insert the text "1a" in the field "Register". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please enter a number." appears on the screen.
4.4	Test	5. (S) Let the field "Register" empty. 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please fill out this field." appears on the screen.
C5	Scenario	1. (S) Scenario C1 2. (S) Channels configuration table with at least 3 rows. 3. (S) In the row with index number 0 (repeat this test later with row 1 and the last row). 4. (S) "Scale" field ("Additional Data" column), in the case of being a parameter of Modbus type or the OneWire type (case not, choose one of the Modbus types or the OneWire type in the selection field).
5.1	Test	5. (S) Insert the text "1a" in the field "Scale". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please enter a number." appears on the screen.

C6	Scenario	<ol style="list-style-type: none"> 1. (S) Scenario C1 2. (S) Channels configuration table with at least 3 rows. 3. (S) In the row with index number 0 (repeat this test later with row 1 and the last row). 4. (S) "Temperature Sensor 1" field ("Additional Data" column), in the case of being a parameter of AVAC type (case not, choose the AVAC type in the selection field).
6.1	Test	<ol style="list-style-type: none"> 5. (S) Insert the number -1 in the field "Temperature Sensor 1". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.
6.2	Test	<ol style="list-style-type: none"> 4. (S) Insert the number 256 in the field "Temperature Sensor 1". 5. (S) Click on the "Configure All" button. 6. (T) Verify that the message "Value must be less than or equal to (N-1)." appears on the screen, where N is the number os channel parameters (check it on this page).
6.3	Test	<ol style="list-style-type: none"> 5. (S) Insert the text "1a" in the field "Temperature Sensor 1". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please enter a number." appears on the screen.
6.4	Test	<ol style="list-style-type: none"> 5. (S) Let the field "Temperature Sensor 1" empty. 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please fill out this field." appears on the screen.
C7	Scenario	<ol style="list-style-type: none"> 1. (S) Scenario C1 2. (S) Channels configuration table with at least 3 rows. 3. (S) In the row with index number 0 (repeat this test later with row 1 and the last row). 4. (S) "Temperature Sensor 2" field ("Additional Data" column), in the case of being a parameter of AVAC type (case not, choose the AVAC type in the selection field).
7.1	Test	<ol style="list-style-type: none"> 5. (S) Insert the number -1 in the field "Temperature Sensor 2". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.
7.2	Test	<ol style="list-style-type: none"> 4. (S) Insert the number 256 in the field "Temperature Sensor 2". 5. (S) Click on the "Configure All" button. 6. (T) Verify that the message "Value must be less than or equal to (N-1)." appears on the screen, where N is the number os channel parameters (check it on this page).
7.3	Test	<ol style="list-style-type: none"> 5. (S) Insert the text "1a" in the field "Temperature Sensor 2". 6. (S) Click on the "Configure All" button. 7. (T) Verify that the message "Please enter a number." appears on the screen.

7.4	Test	<p>5. (S) Let the field "Temperature Sensor 2" empty.</p> <p>6. (S) Click on the "Configure All" button.</p> <p>7. (T) Verify that the message "Please fill out this field." doesn't appear on the screen linked to the field in question.</p>
C8	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) Channels configuration table with at least 3 rows.</p> <p>3. (S) In the row with index number 0 (repeat this test later with row 1 and the last row).</p> <p>4. (S) "Setpoint" field ("Additional Data" column), in the case of being a parameter of AVAC type (case not, choose the AVAC type in the selection field).</p>
8.1	Test	<p>5. (S) Insert the number -1 in the field "Setpoint".</p> <p>6. (S) Click on the "Configure All" button.</p> <p>7. (T) Verify that the message "Value must be greater than or equal to 5." appears on the screen.</p>
8.2	Test	<p>5. (S) Insert the number 256 in the field "Setpoint".</p> <p>6. (S) Click on the "Configure All" button.</p> <p>7. (T) Verify that the message "Value must be less than or equal to 30." appears on the screen.</p>
8.3	Test	<p>5. (S) Insert the text "1a" in the field "Setpoint".</p> <p>6. (S) Click on the "Configure All" button.</p> <p>7. (T) Verify that the message "Please enter a number." appears on the screen.</p>
8.4	Test	<p>5. (S) Let the field "Setpoint" empty.</p> <p>6. (S) Click on the "Configure All" button.</p> <p>7. (T) Verify that the message "Please fill out this field." appears on the screen.</p>
C9	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) Channels configuration table with at least 3 rows.</p> <p>3. (S) In the row with index number 0 (repeat this test later with row 1 and the last row).</p> <p>4. (S) "Type" selection field</p>
9.1	Test	<p>5. (S) The selected value of the selection field of the "Type" is "None" and the "Additional Data" cell is empty.</p> <p>6. (S) Select a Modbus type in the "Type" selection field.</p> <p>7. (T) Verify that input fields appear to custom the selected Modbus type parameter (Mbus Conf#; Register; Format; First Byte/Word; Scale).</p> <p>PS: - If at the step 5 "Type" selection field has not the value "None" selected and the "Additional Data" cell is not empty, try test 9.2 first.</p> <p>- Repeat the test for all the Modbus types.</p>

9.2	Test	<p>5. (S) The selected value of the selection field of the "Type" is one of the Modbus types and the "Additional Data" cell has input fields to custom the selected Modbus type parameter (Mbus Conf#; Register; Format; First Byte/Word; Scale).</p> <p>6. (S) Select "None" in the "Type" selection field.</p> <p>7. (T) Verify that the "Additional Data" cell is empty.</p> <p>PS: - If at the step 5 "Type" selection field has not one of the Modbus types selected and the "Additional Data" cell has not input fields to custom the selected Modbus type parameter (Mbus Conf#; Register; Format; First Byte/Word; Scale), try test 9.1 first. - Repeat the test for all the Modbus type choosing it as needed.</p>
9.3	Test	<p>5. (S) The selected value of the selection field of the "Type" is the OneWire type and the "Additional Data" cell has input fields to custom this type of parameter (Register; Format; First Byte/Word; Scale).</p> <p>6. (S) Select "None" in the "Type" selection field.</p> <p>7. (T) Verify that the "Additional Data" cell is empty.</p> <p>PS: - If at the step 5 "Type" selection field has not the OneWire type selected and the "Additional Data" cell has not input fields to custom this type of parameter (Register; Format; First Byte/Word; Scale), try test 9.5 first.</p>
9.4	Test	<p>5. (S) Select one the Modbus types in the "Type" selection field (different from the already selected, if it is the case, to allow changing).</p> <p>6. (T) Verify that input fields appear to custom the selected Modbus type parameter having the following values - Mbus Conf#: "0"; Register: "65535"; Format: "32bit UInt"; First Byte/Word: "0: MSB/MSW"; Scale: "1". The "ID" field must have the value "0"; the "Acqu. Period" field must have the value "15"; and "Last Values" cell must show "---".</p>
9.5	Test	<p>5. (S) Select the OneWire type in the "Type" selection field (if it is already OneWire type, choose another one, and choose the OneWire type back).</p> <p>6. (T) Verify that input fields appear to custom the selected parameter having the following values - Register: "65535"; Format: "32bit UInt"; First Byte/Word: "0: MSB/MSW"; Scale: "1". The "ID" field must have the value "0"; the "Acqu. Period" field must have the value "15"; and "Last Values" cell must show "---".</p>
9.6	Test	<p>5. (S) Select the AVAC type in the "Type" selection field (if it is already AVAC type, choose another one, and choose the OneWire type back).</p> <p>6. (T) Verify that input fields appear to custom the selected Modbus type parameter having the following values - Temperature Sensor 1: empty; Temperature Sensor 2: empty; Setpoint: "20". The "ID" field must have the value "0"; the "Acqu. Period" field must have the value "15"; and "Last Values" cell must show "---".</p>
C10	Scenario	<p>1. (S) Scenario C1</p> <p>2. (S) Export Channels Configuration to Repository</p>

10.1	Test	
------	------	--

5. Página de Configurações Gerais

#	Type	Description
C1	Scenario	1. (S) Log in (see Scenario C2 in the Session sheet) 2. (S) Click on the "General Configuration" button to go to the page "generalconf.php".
C2	Scenario	1. (S) Scenario C1 2. (S) "Devices Mapping" form 3. (S) For all "Channel Parameter Index" fields
2.1	Test	4. (S) Insert the number -1 in the field "Channel Parameter Index". 5. (S) Click on the "Configure All" button. 6. (T) Verify that the message "Value must be greater than or equal to 0." appears on the screen.
2.2	Test	4. (S) Insert the number 256 in the field "Channel Parameter Index". 5. (S) Click on the "Configure All" button. 6. (T) Verify that the message "Value must be less than or equal to (N-1)." appears on the screen, where N is the number os channel parameters (check it on Channel Configuration page).
2.3	Test	4. (S) Insert the text "1a" in the field "Channel Parameter Index". 5. (S) Click on the "Configure All" button. 6. (T) Verify that the message "Please enter a number." appears on the screen.
2.4	Test	4. (S) Let the field "Channel Parameter Index" empty. 5. (S) Click on the "Configure All" button. 6. (T) Verify that the message "Please fill out this field." appears on the screen.
C3	Scenario	1. (S) Scenario C1 2. (S) "Devices Mapping" form
3.1	Test	3. (S) Let "Channel Parameter Index" fields with repeated values. 4. (S) Click on the "Configure All" button. 5. (T) Verify that the messages "SUCCESS: General configuration set with success." and "WARNING: There are repeated index values." appear on the screen.
3.2	Test	3. (S) Let "Channel Parameter Index" fields with repeated values. 4. (S) Click on the "Configure All" button. 5. (S) Refresh the page. 6. (T) Verify that the message "WARNING: There are repeated index values." appears on the

		screen.
3.3	Test	<ol style="list-style-type: none">3. (S) Let "Channel Parameter Index" fields without repeated values.4. (S) Click on the "Configure All" button.5. (T) Verify that the messages "SUCCESS: General configuration set with success." appears on the screen and "WARNING: There are repeated index values." doesn't appear on the screen.
3.4	Test	<ol style="list-style-type: none">3. (S) Let "Channel Parameter Index" fields without repeated values.4. (S) Click on the "Configure All" button.5. (S) Refresh the page.6. (T) Verify that the message "WARNING: There are repeated index values." doesn't appear on the screen.

F. Anexo F - Proposta de API do iHub

Proposta de API do iHub

INTEGRAÇÃO NO IMOD

1 Introdução

No âmbito do projeto de integração do iMod, tem sido criado uma interface Web para configurar a comunicação e envio/recolha de dados de um dado dispositivo iMod. Nesse sentido, está a ser integrado o protocolo iHub para envio/recolha de dados para um servidor remoto, mas a recolha entre a interface Web e o protocolo iHub tem ser estabelecida e, de preferência, de uma forma elegante, eficiente e intuitiva.

No seguimento dessa ideia, decidi criar uma API para o iHub, ou seja, uma interface lógica para o protocolo iHub que atende aos pedidos da interface Web do iMod para os efeitos pretendidos.

2 Proposta

Aqui é apresentado um esboço do que se pretende implementar e de alguns aspetos e detalhes que serão importantes para a criação e bom funcionamento da API.

A ligação entre a interface Web e API é feita através de *sockets* TCP/IP, em que são enviados pacotes em formato JSON, correspondendo aos pedidos e respostas das transições.

A conexão entre ambas as partes não é constante, pelo que é feita numa só vez, aquando a transição.

A API fica à espera de pedidos, podendo atender apenas a um pedido de cada vez, isto é, não terá servidores dedicados para permitir o atendimento de vários clientes em simultâneo.

2.1 Request (pedido)

Aqui são apresentadas as combinações de código-função que serão associadas a um pedido por parte da interface Web.

ID Função

0. `server_conf getServerConf();`//Obtém configuração de servidor remoto associado ao iHub. Devolve uma estrutura `server_conf`.
1. `int setServerConfTime(unsigned int year, unsigned int month, unsigned int day, unsigned int hours, unsigned int minutes, unsigned int seconds);`//Configura uma nova data no servidor remoto associado ao iHub. Poderá usar “`struct server_conf_time`” no lugar como argumento. Retorna valor correspondente à forma como decorreu a operação.
2. `int connectServer(char[] host, unsigned int port);`//Testa conexão a um *host* remoto e porto. Retorna valor correspondente à forma como decorreu a operação.
3. `int setServerConf(unsigned int serial, unsigned int as_tcp_server, char host[], unsigned int port, unsigned int comm_period, unsigned int encap_crc, unsigned int errors_before_reboot);`//Configura dados principais de configuração de servidor remote associado ao iHub. Poderá usar “`struct server_conf_server`” no lugar como argumento. Retorna valor correspondente à forma como decorreu a operação.

4. `channel_params_conf getChannelsConf();`//Obtém configuração de canais de parâmetros. Devolve uma estrutura `channel_params_conf`.
5. `int setChannelsConf(channel_param_conf params[]);`//Configura canais de parâmetros. Retorna valor correspondente à forma como decorreu a operação.
6. `int setChannelConf(channel_param_conf param, unsigned int index);` //Configura um canal de parâmetro com um índice especificado. Retorna valor correspondente à forma como decorreu a operação.

2.2 Response (resposta)

Aqui são apresentadas os códigos de retorno que serão associados a uma resposta para a interface Web.

ID Representação

0. SUCCESS_RESPONSE - Sucesso
1. JSON_FAIL_RESPONSE - Má formatação de formato JSON de pedido ou de resposta (e por não corresponder aos valores, tipos de valores e/ou estrutura esperada segundo a API).
2. FAIL_RESPONSE - Falha da API ou operação apresenta resultado negativo (por exemplo, o teste de conexão a servidor falhou).
3. TECH_ISSUE_RESPONSE – Falha técnica (por exemplo, não conseguiu criar um *socket* para ligar a servidor no teste de conexão).

2.3 Estruturas (keys) JSON envolvidas nos pedidos e respostas

2.3.1 Request

- 'requestcode' – Código do pedido para corresponder à função que se pretende

2.3.1.1 ID 0 – getServerConf

Este pedido não necessita de dados adicionais.

2.3.1.2 ID 1 - setServerConfTime

- 'year' – Ano a configurar
- 'month' – Mês a configurar
- 'day' – Dia a configurar

- 'hours' – Horas a configurar
- 'minutes' – Minutos a configurar
- 'seconds' - Segundos a configurar

2.3.1.3 ID 2 - connectServer

- 'testhost' – *Host* para testar conexão
- 'testport' – Porto para testar conexão

2.3.1.4 ID 3 - setServerConf

- 'serial' - N° de série a configurar
- 'astcp' – Configura se é ou não servidor TCP
- 'host' – *Host* a configurar
- 'port' – Porto a configurar
- 'period' – Período de comunicação a configurar
- 'crc' – Configura se encapsula ou não em CRC
- 'nerrors' – N° de erros para reiniciar a ser configurado

2.3.1.5 ID 4 - getChannelsConf

Este pedido não necessita de dados adicionais.

2.3.1.6 ID 5 - setChannelsConf

- Um *array* de dados, em que cada item tem as seguintes chaves (cada item identificado pelo índice: de 0 a N-1, em que N é o n° de itens):
 - 'type' – Tipo de parâmetro a ser configurado
 - 'id' – ID do parâmetro a ser configurado
 - 'period' – Período de aquisição de parâmetro a ser configurado

- 'mbusconf' – Configuração Modbus a ser configurada (valor obrigatório para parâmetro de um dos tipos de Modbus)
- 'mregister' – Registo Modbus a ser configurado (valor obrigatório para parâmetro de um dos tipos de Modbus)
- 'mformat' – Formato Modbus a ser configurado (valor obrigatório para parâmetro de um dos tipos de Modbus)
- 'mfirstbw' – Primeiro *byte/word* Modbus a ser configurado (valor obrigatório para parâmetro de um dos tipos de Modbus)
- 'mscale' – Escala Modbus a ser configurada (valor obrigatório para parâmetro de um dos tipos de Modbus)

2.3.1.7 ID 6 - setChannelConf

- Um *array* de dados tem como chave o valor do índice do parâmetro a configurar, sendo os dados os seguintes:
 - 'type' – Tipo de parâmetro a ser configurado
 - 'id' – ID do parâmetro a ser configurado
 - 'period' – Período de aquisição de parâmetro a ser configurado
 - 'mbusconf' – Configuração Modbus a ser configurada (valor obrigatório para parâmetro de um dos tipos de Modbus)
 - 'mregister' – Registo Modbus a ser configurado (valor obrigatório para parâmetro de um dos tipos de Modbus)
 - 'mformat' – Formato Modbus a ser configurado (valor obrigatório para parâmetro de um dos tipos de Modbus)
 - 'mfirstbw' – Primeiro *byte/word* Modbus a ser configurado (valor obrigatório para parâmetro de um dos tipos de Modbus)
 - 'mscale' – Escala Modbus a ser configurada (valor obrigatório para parâmetro de um dos tipos de Modbus)

2.3.2 Response

- 'responsecode' – Código de resposta para saber como correu a execução do pedido
- 'responsemsg' – Mensagem associada à resposta para efeito informativo

2.3.2.1 ID 0 – `getServerConf`

- 'year' - Ano
- 'month' - Mês
- 'day' - Dia
- 'hours' - Horas
- 'minutes' - Minutos
- 'seconds' – Segundos
- 'macid' – MAC ID
- 'serial' - N° de série
- 'astcp' - Se é ou não servidor TCP
- 'host' - *Host*
- 'port' - Porto
- 'period' – Período de comunicação
- 'crc' - Se encapsula ou não em CRC
- 'nerrors' - N° de erros para reiniciar

2.3.2.2 ID 1 - `setServerConfTime`

Esta resposta não necessita de dados adicionais.

2.3.2.3 ID 2 - `connectServer`

Esta resposta não necessita de dados adicionais.

2.3.2.4 ID 3 – setServerConf

Esta resposta não necessita de dados adicionais.

2.3.2.5 ID 4 – getChannelsConf

- 'year' - Ano
- 'month' - Mês
- 'day' - Dia
- 'hours' - Horas
- 'minutes' - Minutos
- 'seconds' – Segundos
- Um *array* de dados, em que cada item tem as seguintes chaves (cada item identificado pelo índice: de 0 a N-1, em que N é o n° de itens):
 - 'type' – Tipo de parâmetro
 - 'id' – ID do parâmetro
 - 'period' – Período de aquisição do parâmetro
 - 'mbusconf' – Configuração Modbus (valor obrigatório para parâmetro de um dos tipos de Modbus)
 - 'mregister' – Registo Modbus (valor obrigatório para parâmetro de um dos tipos de Modbus)
 - 'mformat' – Formato Modbus (valor obrigatório para parâmetro de um dos tipos de Modbus)
 - 'mfirstbw' – Primeiro *byte/word* Modbus (valor obrigatório para parâmetro de um dos tipos de Modbus)
 - 'mscale' – Escala Modbus (valor obrigatório para parâmetro de um dos tipos de Modbus)

- 'year' – Ano (da data/hora da última recolha de valor do parâmetro)
- 'month' – Mês (da data/hora da última recolha de valor do parâmetro)
- 'day' – Dia (da data/hora da última recolha de valor do parâmetro)
- 'hours' – Horas (da data/hora da última recolha de valor do parâmetro)
- 'minutes' – Minutos (da data/hora da última recolha de valor do parâmetro)
- 'seconds' – Segundos (da data/hora da última recolha de valor do parâmetro)
- 'lastvalue' – Último valor recolhido do parâmetro

2.3.2.6 ID 5 – setChannelsConf

Esta resposta não necessita de dados adicionais.

2.3.2.7 ID 6 – setChannelConf

Esta resposta não necessita de dados adicionais.

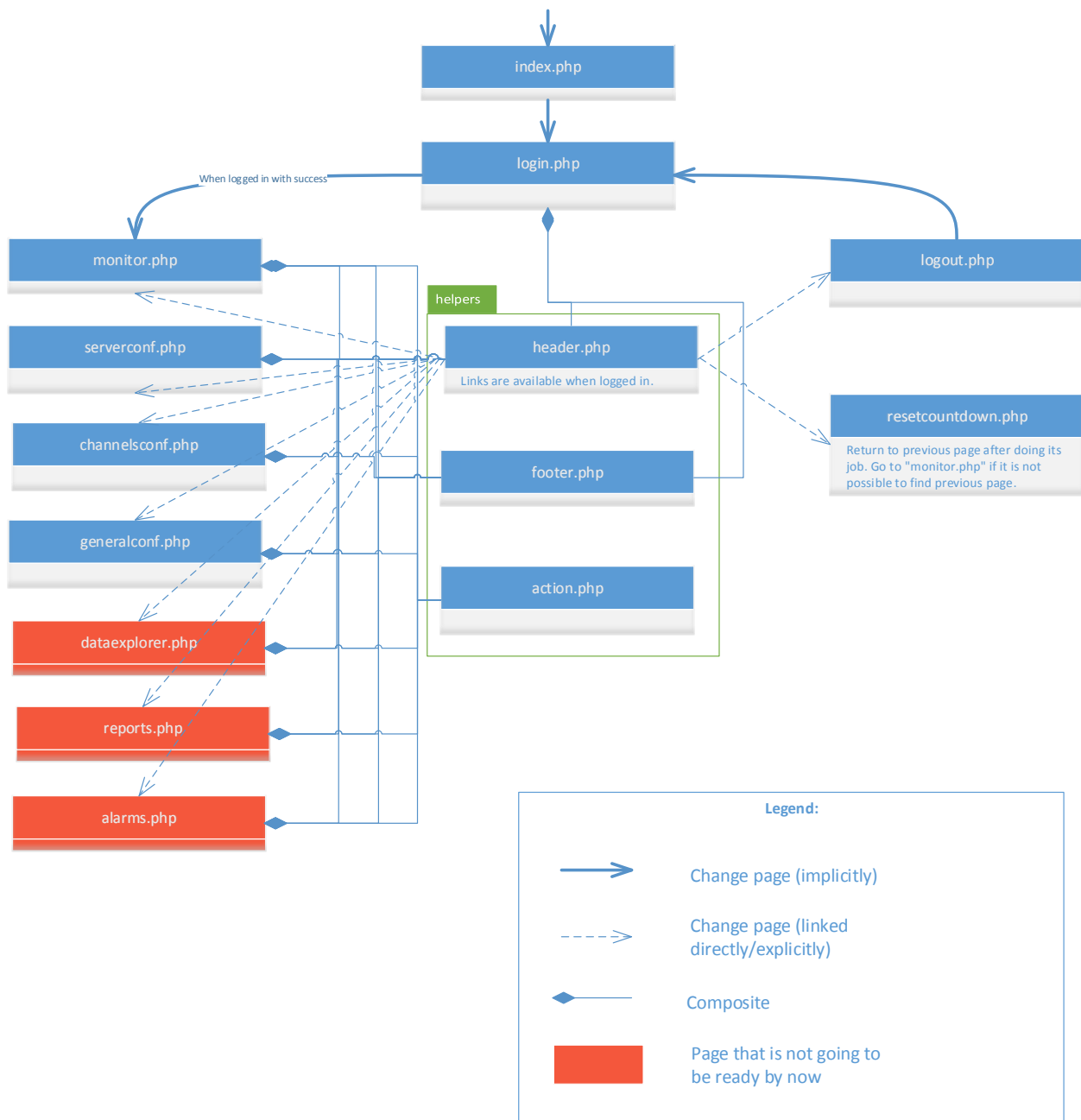
3 Conclusão

No seguimento desta proposta, pretende-se discutir e detalhar a sua forma e requisitos de implementação.

Foi elaborado um esqueleto da aplicação, que ajudou a corrigir e detalhar esta proposta.

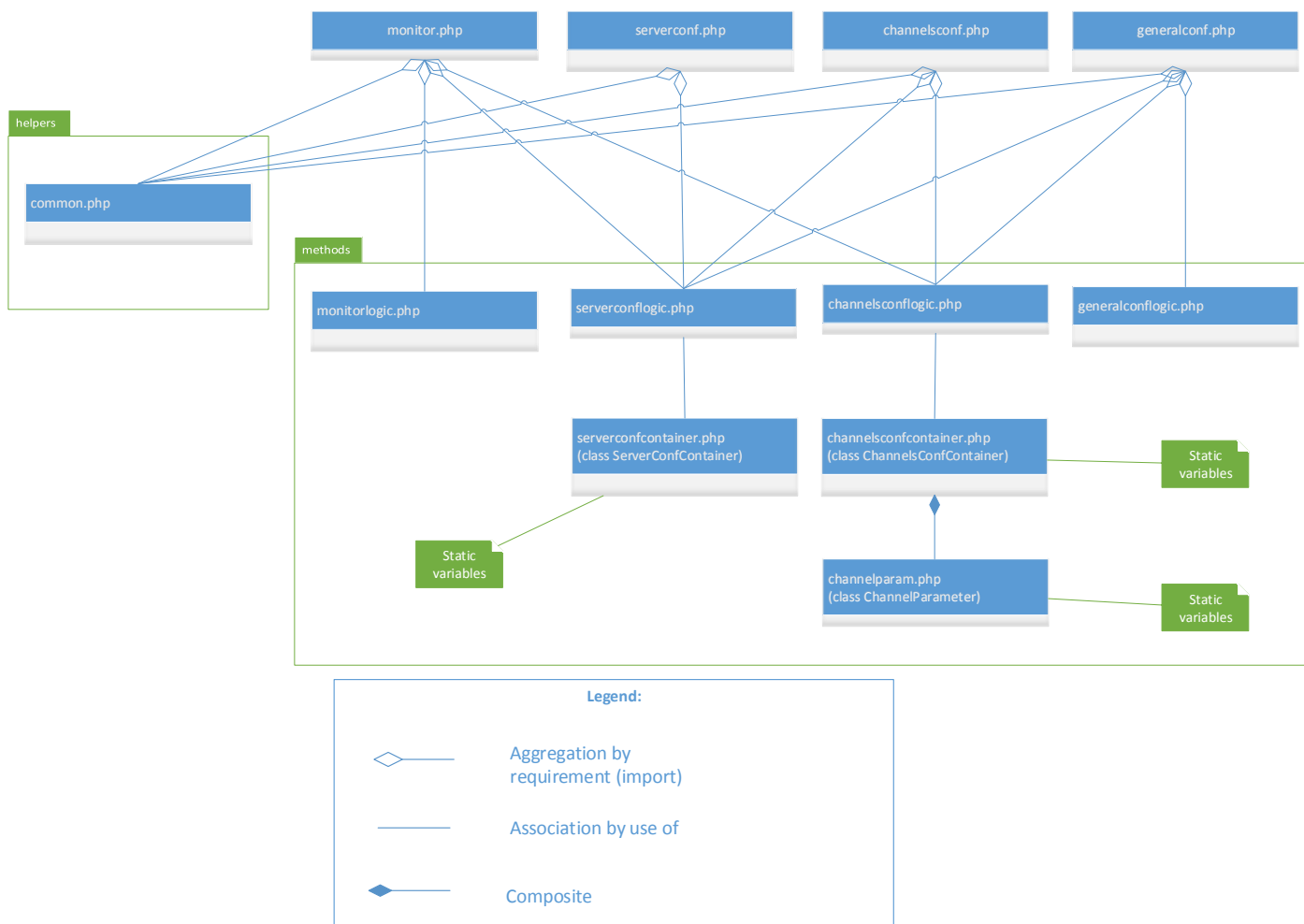
G. Anexo G - Arquitetura de interface Web do iMod

1. Interface de utilizador



- When opening the project from the root, it always start in “index.php”. “index.php” redirects to “login.php” (login page).
- If it is already logged in or when the login is done with success, it is redirected to “monitor.php”. “login.php” is composed by “header.php” (header page subpart) and “footer.php” (footer page subpart).
- “header.php” has links to redirect to pages, but only when logged in. It includes “resetcountdown.php” (to reset the user session timer) and “logout.php” (to log out from user session).
- “logout.php” ends user session and redirect to “login.php” page.
- The pages “monitor.php” (Monitor page), “serverconf.php” (Server Configuration page), “channelsconf.php” (Channels Configuration page), “generalconf.php” (General Configuration page), “dataexplorer.php” (Data Explorer page), “reports.php” (Reports page) and “alarms.php” (Alarms page) are composed by “header.php”, “action.php” (page sub-part that shows information after user action, as for example, submitting a configuration) and “header.php”.

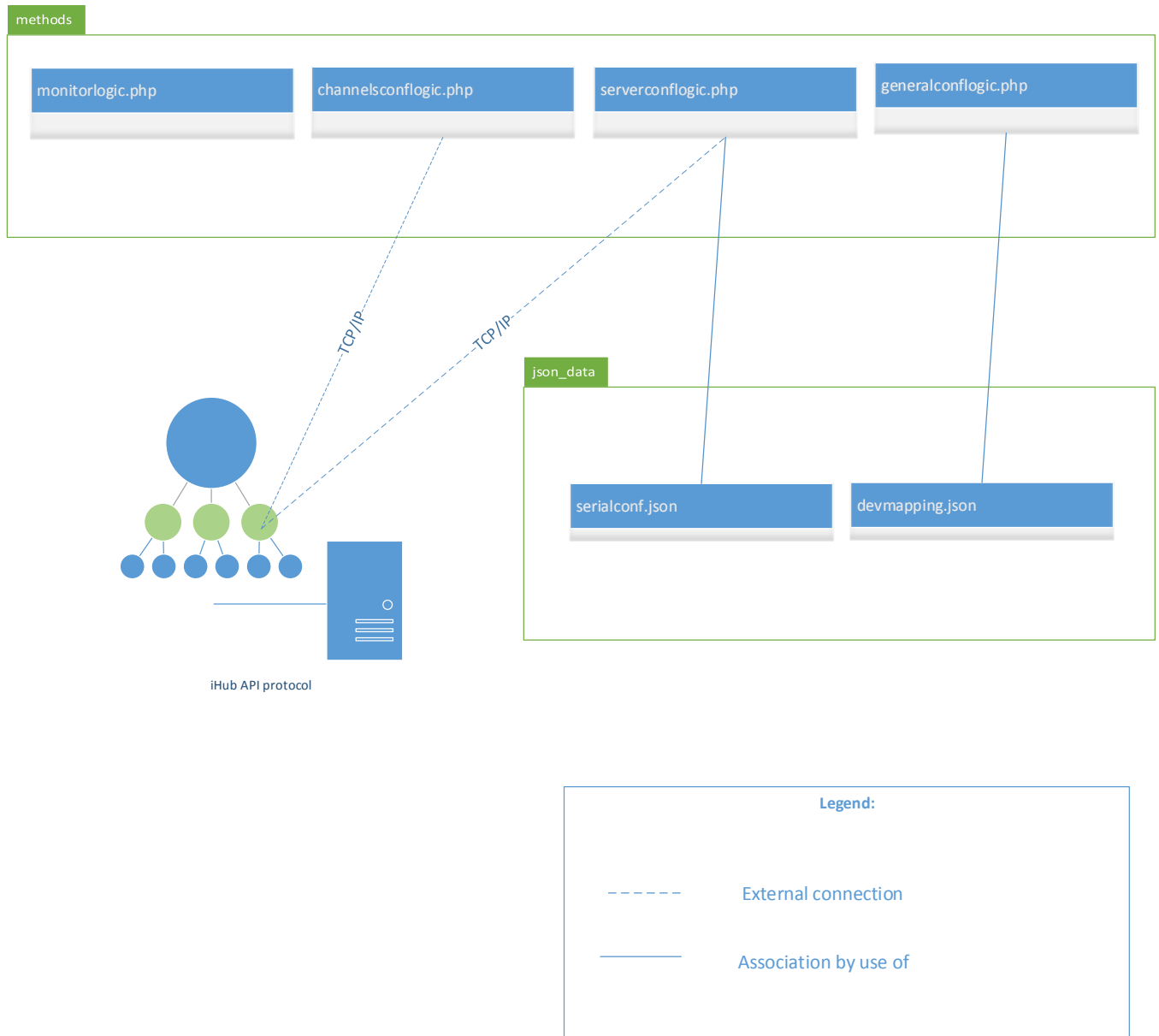
2. Lógica



- The pages above require files that contain functions, containers and all the main logical part: “common.php” (contains generic functions) in the “helpers” folder; and other ones in the “methods” folder, as it can be seen in the diagram.
- The “serverconflogic.php” file uses functions from “serverconfcontainer.php”.
- The “channelsconflogic.php” file uses functions from “channelsconfcontainer.php”.
- The container files (“serverconfcontainer.php” - represents a server’s configuration - and “channelsconfcontainer.php” - represents a channel parameters’ configuration) and container related (“channelparam.php” - represents a channel parameter) have classes which contains static variables, in order to have one unique instance of each available.

- The class ChannelsConfContainer (“channelsconfcontainer.php”) is composed by many instances of ChannelParameter (“channelparam.php”), that are an array of channel parameters from the whole configuration.

3. Persistência e Serviços



- In the folder "json_data" there are files in JSON format with data for persistence, which are used by some logical files.
- Some logical files obtain and configure data in the iHub API protocol through a TCP/IP connection.

H. Anexo H - Manual de utilizador da interface Web do iMod

Interface Web do iMod

MANUAL DE UTILIZADOR

1. Introdução

No âmbito do projeto de integração do dispositivo iMod, em parceria com o projeto do Banco Popular, foi realizado uma interface Web em PHP a fim de poder configurar e obter valores na ordem do contexto dos requisitos definidos.

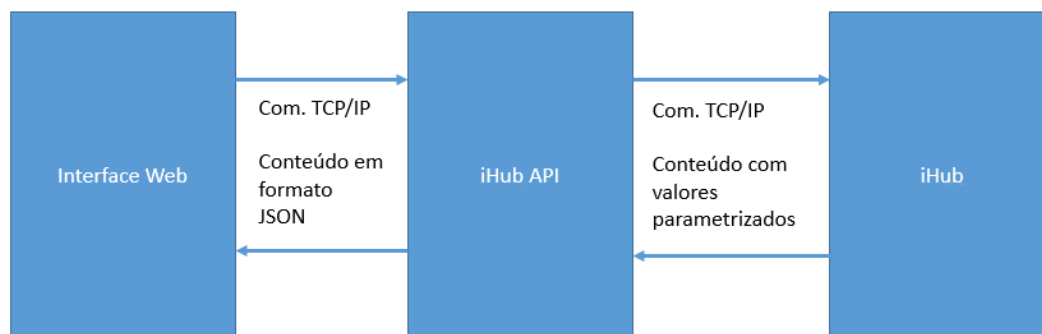
Este documento apresenta uma abordagem genérica da envolvente do projeto, seguido com detalhes da configuração e utilização da interface Web em questão.

2. Abordagem geral

Esta interface permite obter e configurar um protocolo da ISA chamado iHub, que está em remodelação, mas já permite realizar configurações básicas do mesmo. Nesse sentido, foi também realizada uma API, de modo em que esta é uma fachada que permite obter e alterar configurações do protocolo iHub de forma intuitiva e através de transações (pedido e resposta) via TCP/IP com dados no formato JSON.

É possível entender genericamente o processo das entidades envolvidas neste processo, olhando para o esquema abaixo:

Integração iMod

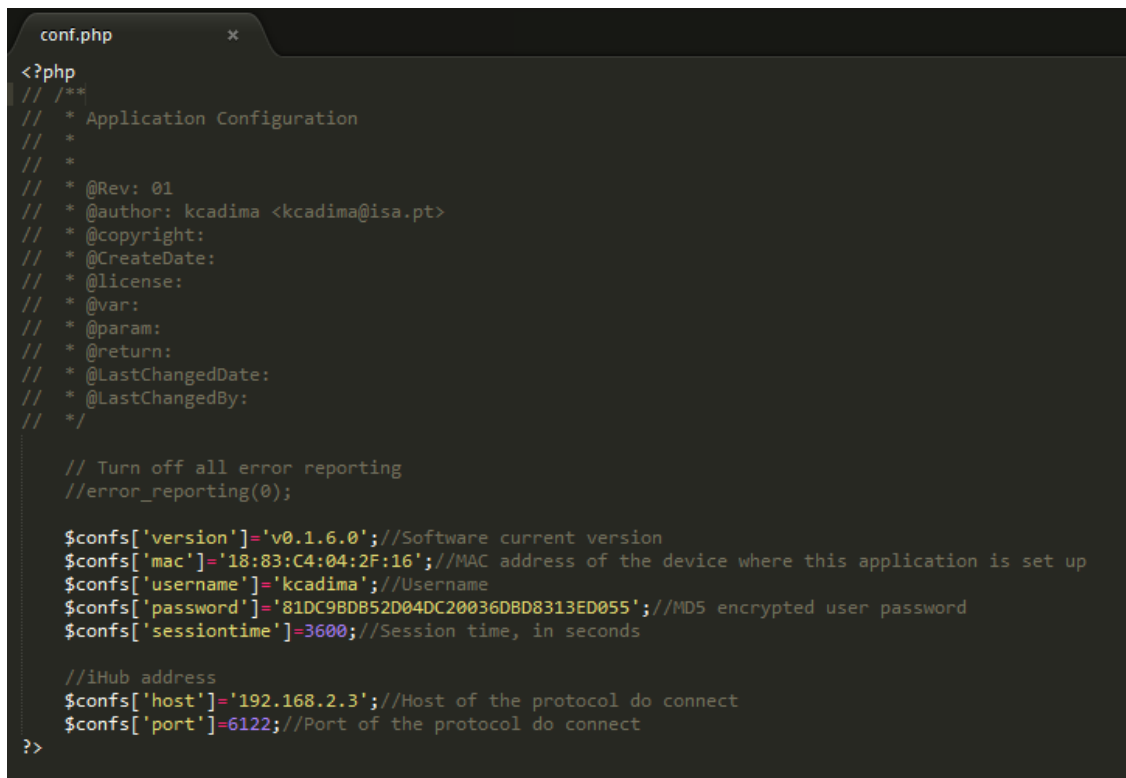


3. Configuração de interface Web

A pasta do projeto Web de nome “imodinterface” tem de se encontrar na pasta “htdocs” do Apache. O Apache tem de estar em execução para poder carregar as páginas Web.

Dentro da pasta do projeto existe uma pasta “conf”, que contém um ficheiro chamado “conf.php”, de configuração para a interface Web em questão (como se pode ver na imagem abaixo).

A configuração contém a versão desta aplicação; o endereço MAC do dispositivo em que se encontra (MAC do iMod, neste sentido); as credenciais de acesso à conta de utilizador (nome de utilizar e palavra-passe encriptada em MD5); o tempo de sessão de utilizador em segundos; e o *host* e porto, que se referem à API do iHub.



```
conf.php x
<?php
// **
// * Application Configuration
// *
// *
// * @Rev: 01
// * @author: kcadima <kcadima@isa.pt>
// * @copyright:
// * @CreateDate:
// * @license:
// * @var:
// * @param:
// * @return:
// * @LastChangedDate:
// * @LastChangedBy:
// */

// Turn off all error reporting
//error_reporting(0);

$configs['version']='v0.1.6.0';//Software current version
$configs['mac']='18:83:C4:04:2F:16';//MAC address of the device where this application is set up
$configs['username']='kcadima';//Username
$configs['password']='81DC98DB52D04DC200360BD8313ED055';//MD5 encrypted user password
$configs['sessiontime']=3600;//Session time, in seconds

//iHub address
$configs['host']='192.168.2.3';//Host of the protocol do connect
$configs['port']=6122;//Port of the protocol do connect
?>
```

Figura 37 - Ficheiro de configuração de interface Web

4 Utilização

4.1 Login

Para aceder à sessão de utilizador é necessário introduzir as credenciais definidas para o utilizador (no ficheiro de configuração referido no tópico anterior).

Quando a sessão de utilizador é iniciada com sucesso, é se redirecionado para a página *Monitor*, sendo possível navegar pelas várias páginas existentes através dos botões e ligações existentes no cabeçalho para o efeito.

A sessão de utilizador é encerrada após o tempo de sessão visível no ecrã chegar a zero segundos. No entanto é possível terminá-la explicitamente clicando em “*Log out*” ou reiniciar o tempo de sessão clicando em “*Restart Session Countdown*”.

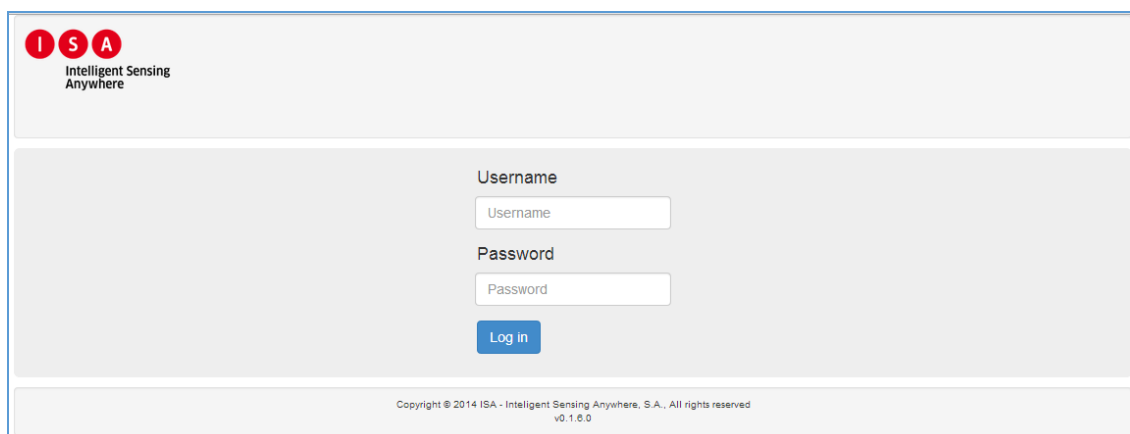
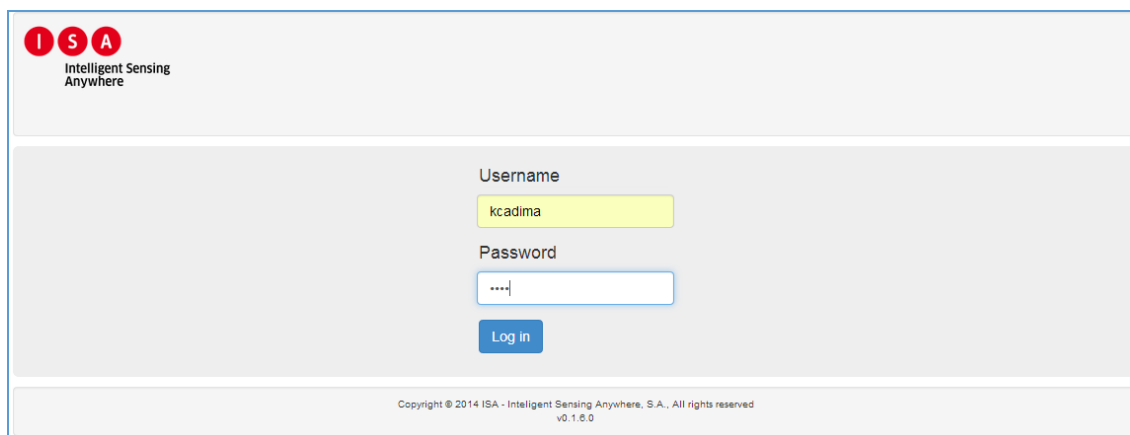
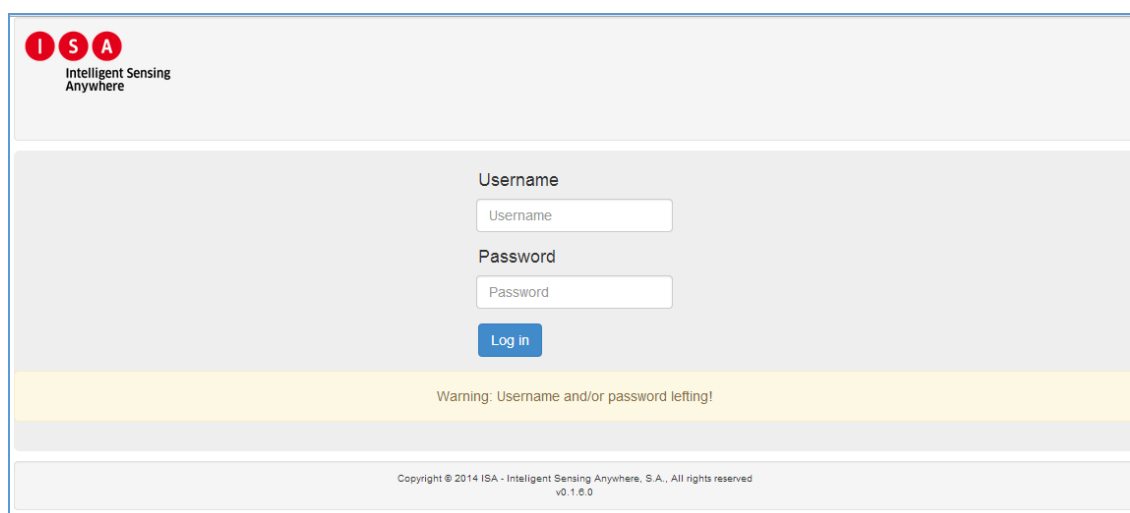


Figura 38 – Esta é a página de *login*. É apresentada quando a sessão de utilizador não se encontra aberta.



The screenshot shows the ISA login interface. At the top left is the ISA logo with the text "Intelligent Sensing Anywhere". Below the logo is a form with two input fields: "Username" containing the text "kcadima" and "Password" containing four dots. A blue "Log In" button is positioned below the password field. At the bottom of the page, there is a small copyright notice: "Copyright © 2014 ISA - Intelligent Sensing Anywhere, S.A., All rights reserved v0.1.0.0".

Figura 39 – Para fazer login é necessário introduzir o nome de utilizador e a palavra-passe nos respetivos campos, clicando no botão “Login” para validar a operação.



This screenshot shows the same ISA login interface as Figure 39, but with a yellow warning banner at the bottom of the form area. The banner contains the text "Warning: Username and/or password lefting!". The "Username" and "Password" fields are now empty. The "Log In" button remains visible. The copyright notice at the bottom is the same as in Figure 39.

Figura 40 – É mostrado o aviso “Warning: Username and/or password lefting!” quando um dos campos das credenciais está vazio após a submissão de login. Quando pelo menos uma das credenciais não está correta, é mostrado o aviso “Warning: Wrong username and/or password!”.

4.2 Mensagens de interface

Nas páginas Web, aquando sessão de utilizador ativa, são apresentadas mensagens de interface, ou seja, quando por exemplo o utilizador submete uma configuração clicando no botão para esse efeito.

Existem 4 tipos de mensagem de interface:

- **INFORMAÇÃO** – Mensagens de âmbito informativo, tais como a informação de espera de uma operação após submissão pelo utilizador. Tem a cor azul.
- **SUCESSO (SUCCESS)** – Mensagem de sucesso de operação, como por exemplo para uma configuração efetuada com sucesso. Tem a cor verde.
- **AVISO (WARNING)** – Mensagem de aviso, que serve para avisar que um dado pormenor pode não estar em conformidade ou que uma operação não foi efetuada com sucesso, mas não tem gravidade. Tem a cor amarela.
- **ERRO (ERROR)** – Mensagem de erro, que está associada a mensagem de maior gravidade, como por exemplo o facto de não ser estabelecida conexão com o protocolo iHub para obter dados. Tem a cor vermelha.

4.3 Monitor

Esta página permite ver informações genéricas do sistema; os valores dos dispositivos que se encontram nas instalações (mapeados na página *General Configuration*); e permite exportar todas as configurações e *logs* do sistema para um repositório remoto (funcionalidade não implementada).

The screenshot shows the ISA Monitor interface. At the top left is the ISA logo (Intelligent Sensing Anywhere). The top right shows the user 'kcadima' with a session expiration of 49 minutes and links for 'Restart Session Countdown', 'General Configuration', and 'Log out'. Below this is a navigation menu with buttons for 'Monitor', 'Server Configuration', 'Channels Configuration', 'Data Explorer', 'Reports', and 'Alarms'. The main content area is titled 'Monitor' and contains a 'System Info' section with details like 'Unit Time: 16/06/2014 14:50' and 'Connection State: Implementation State'. Below that is a 'Devices' table with 4 rows and 3 columns: '#', 'Type', and 'Value'. The table shows values for AVAC, Lighting, Advertisements, and AVAC Setpoint. A blue button 'Export Configurations and Logs to Repository' is located below the table. The footer contains copyright information for ISA.

#	Type	Value
0	AVAC - Turned On	-
1	Lighting - Turned On	Yes
2	Advertisements - Turned On	Yes
3	AVAC - Setpoint (°C)	3.2

Figura 41 – Página *Monitor* com valores de dispositivos monitorizados e um botão para exportação de configurações e logs.

This screenshot shows the same ISA Monitor interface but with an error message. The error message is displayed in a red box: 'ERROR: Can't get channels configuration. Failures: No connection to the protocol - host: 172.17.0.160 - port: 6122'. The 'System Info' section is identical to the previous screenshot. The 'Devices' table now shows 'NOT CONNECTED' for all values. The 'Export Configurations and Logs to Repository' button is still present. The footer contains the same copyright information.

#	Type	Value
0	AVAC - Turned On	NOT CONNECTED
1	Lighting - Turned On	NOT CONNECTED
2	Advertisements - Turned On	NOT CONNECTED
3	AVAC - Setpoint (°C)	NOT CONNECTED

Figura 42 – Neste caso, não foi possível obter valores por não se ter conseguido conectar com o protocolo iHub a fim de obter valores dos parâmetros mapeados.

4.4 Server Configuration

Nesta página é possível configurar as configurações genéricas do protocolo iHub, as portas série e exportar estas configurações para um repositório remoto (funcionalidade não implementada).

The screenshot shows the 'Server Configuration' page of the iMod web interface. At the top left is the 'ISA Intelligent Sensing Anywhere' logo. The top right shows the user 'kcadima' with a session expiration of 45 minutes and links for 'Restart Session Countdown', 'General Configuration', and 'Log out'. A navigation bar contains buttons for 'Monitor', 'Server Configuration', 'Channels Configuration', 'Data Explorer', 'Reports', and 'Alarms'. The main content area is titled 'Server Configuration' and includes: 'Unit Time: 2014/06/16 02:22:42' and 'Local time: 2014/06/16 14:22:34' with a 'Set Time' button; a 'Test Connection:' section with a '(Remote Host:Port)/(Local Port):' field and a '12345' field, plus a 'Connect Now' button; a 'Unit Configuration:' section with a '(MAC: 00:30:05:EA:B8:29)' label and a 'Configure Unit' button; a 'Serial:' field with the value '129'; a 'Server Communication:' section with 'Function as a TCP server:' checked, a '(Remote Host:Port)/(Local Port):' field, and a 'Communication Period:' field with the value '65520' and 'minutes' label; a 'Protocol Framing:' section with 'Encapsulate protocol in frame with CRC:' unchecked; and an 'Errors before reboot (0: disable reboot):' field with the value '12'. At the bottom, there is a 'Serial Port (RS485) Configuration:' section with a 'Configure RS485' button.

Figura 43 – Página *Server Configuration*, é visível a configuração do tempo do servidor, um teste de conexão a servidor e a configuração geral do servidor.

ISA Intelligent Sensing Anywhere

Username: kcadima
Session expires in 42 minutes

Restart Session Countdown
General Configuration
Log out

Monitor Server Configuration Channels Configuration Data Explorer Reports Alarms

Server Configuration

Waiting... ⌛

Unit Time: 2014/06/16 02:19:25

Local time: 2014/06/16 14:25:30 [Set Time](#)

Test Connection:

(Remote Host:Port)/(Local Port): : 12345 [Connect Now](#)

Unit Configuration: (MAC: 00:30:05:EA:B8:29) [Configure Unit](#)

Serial: 129

Server Communication:

Function as a TCP server:

(Remote Host:Port)/(Local Port): : 12345

Communication Period: 65520 minutes

Protocol Framing:

Encapsulate protocol in frame with CRC:

Errors before reboot (0: disable reboot): 12

Figura 44 – Ao clicar num dos botões de submissão para efetuar uma operação, é mostrado um painel informativo indicando para esperar.

ISA Intelligent Sensing Anywhere

Username: kcadima
Session expires in 42 minutes

Restart Session Countdown
General Configuration
Log out

Monitor Server Configuration Channels Configuration Data Explorer Reports Alarms

Server Configuration

SUCCESS: Unit time configured with success.

Unit Time: 2014/06/16 14:26:10

Local time: 2014/06/16 14:26:11 [Set Time](#)

Test Connection:

(Remote Host:Port)/(Local Port): : 12345 [Connect Now](#)

Unit Configuration: (MAC: 00:30:05:EA:B8:29) [Configure Unit](#)

Serial: 129

Server Communication:

Function as a TCP server:

(Remote Host:Port)/(Local Port): : 12345

Communication Period: 65520 minutes

Protocol Framing:

Encapsulate protocol in frame with CRC:

Errors before reboot (0: disable reboot): 12

Figura 45 – É mostrado o painel de sucesso de operação, que neste caso corresponde à configuração do tempo de servidor.

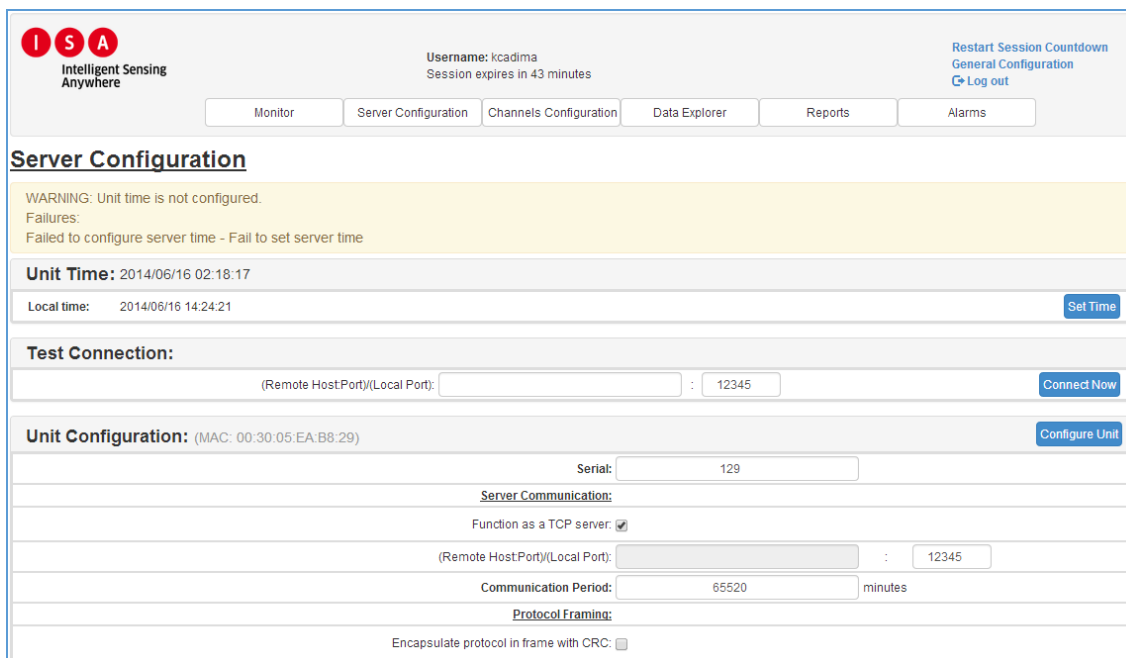


Figura 46 – É mostrado o painel de aviso, que indica que não foi possível efetuar a configuração do tempo do servidor.

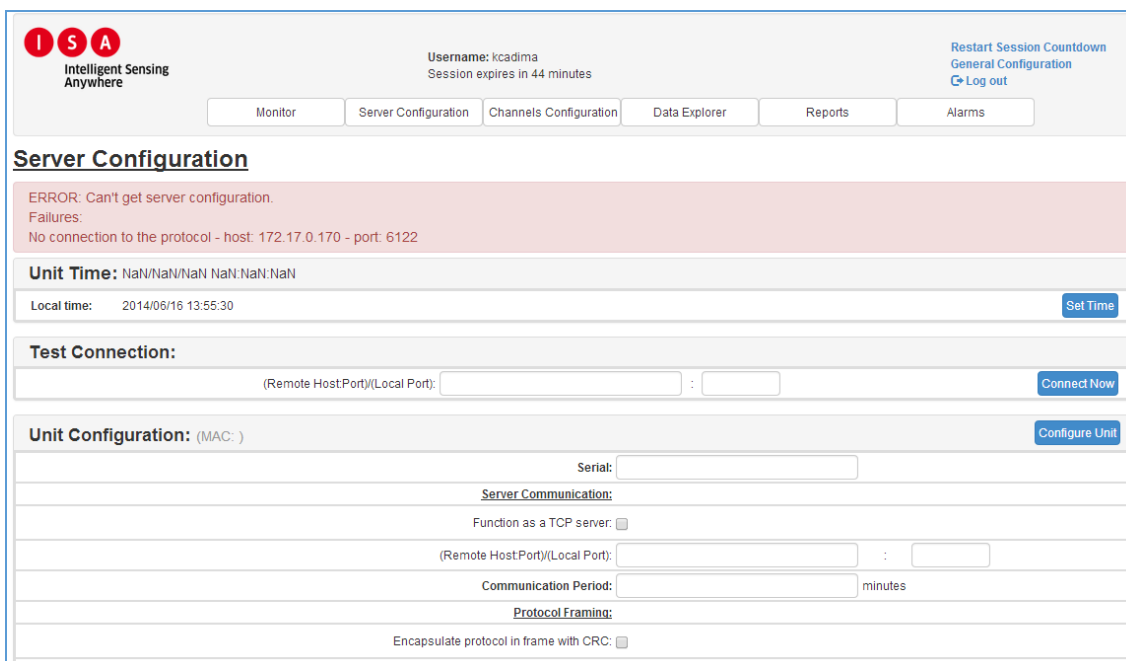


Figura 47 - Neste caso, não foi possível obter valores de configuração principais do protocolo iHub.

Protocol Framing:
 Encapsulate protocol in frame with CRC:
 Errors before reboot (0: disable reboot):

Serial Port (RS485) Configuration: Configure RS485

	#	Port	Baud Rate	Data Bits	Stop Bits	Parity	Resp. Timeout	Query Delay	Retries
Current configurations:	0	ttyS0	3000	8	2	even	12 ms	2 ms	1
	1	ttyS0	2500	8	1	even	22 ms	12 ms	62
	2	ttyS1	2400	8	2	none	12 ms	34 ms	30
	3	ttyS2	19200	8	2	even	44 ms	4 ms	33
New configurations:	0	<input type="text" value="ttyS0"/>	<input type="text" value="Custom"/> <input type="text" value="3000"/>	8	<input type="text" value="2"/>	<input type="text" value="even"/>	<input type="text" value="12"/> ms	<input type="text" value="2"/> ms	<input type="text" value="1"/>
	1	<input type="text" value="ttyS0"/>	<input type="text" value="Custom"/> <input type="text" value="2500"/>	8	<input type="text" value="1"/>	<input type="text" value="even"/>	<input type="text" value="22"/> ms	<input type="text" value="12"/> ms	<input type="text" value="62"/>
	2	<input type="text" value="ttyS1"/>	<input type="text" value="2400"/> <input type="text" value="2400"/>	8	<input type="text" value="2"/>	<input type="text" value="none"/>	<input type="text" value="12"/> ms	<input type="text" value="34"/> ms	<input type="text" value="30"/>
	3	<input type="text" value="ttyS2"/>	<input type="text" value="19200"/> <input type="text" value="19200"/>	8	<input type="text" value="2"/>	<input type="text" value="even"/>	<input type="text" value="44"/> ms	<input type="text" value="4"/> ms	<input type="text" value="33"/>

Export Server Configuration to Repository

Copyright © 2014 ISA - Intelligent Sensing Anywhere, S.A., All rights reserved
 v0.1.0.0

Figura 48 – Configuração de portas série e exportação de configuração

4.5 Channels Configuration

Esta página serve para configurar os parâmetros de canais de comunicação, que se encontram no protocolo iHub, podendo-se exportar esta configuração para um repositório remoto (funcionalidade não implementada).

Intelligent Sensing Anywhere

Username: kcadima
Session expires in 41 minutes

Restart Session Countdown
General Configuration
Log out

Monitor Server Configuration Channels Configuration Data Explorer Reports Alarms

Channels Configuration

Configure All

Index	Type	ID	Acqu. Period	Adicional Data	Last Values
0	None	3	15 min		---
1	Modbus FC2	4	11 min	MBus Conf #: 0 Register: 1989 Format: 16bit FP (IEEE754 half) First Byte/Word: 2: LSB/MSW Scale: 1.11111E+29	Time: 2015/01/04 08:04:03 Value: 2.2
2	Modbus FC6	1	15 min	MBus Conf #: 0 Register: 65535 Format: 32bit UInt First Byte/Word: 0: MSB/MSW Scale: 1	Time: 2013/03/06 10:06:05 Value: 6.3
3	OneWire	5	20 min	Register: 1994 Format: 16bit UInt First Byte/Word: 3: LSB/LSW Scale: 2.123	Time: 2016/02/05 09:05:04 Value: 3.2

Figura 49 – Página *Channels Configuration*, onde possível ver uma tabela com os parâmetros a configurar, podendo-se escolher os seus tipos e editar vários atributos dos respetivos parâmetros.

Configure All

Index	Type	ID	Acqu. Period	Adicional Data	Last Values
0	None	3	15 min		---
1	Modbus FC2	4	11 min	MBus Conf #: 0 Register: 1989 Format: 16bit FP (IEEE754 half) First Byte/Word: 2: LSB/MSW Scale: 1.11111E+29	Time: 2015/01/04 08:04:03 Value: 2.2
2	Modbus FC6	1	15 min	MBus Conf #: 0 Register: 65535 Format: 32bit UInt First Byte/Word: 0: MSB/MSW Scale: 1	Time: 2013/03/06 10:06:05 Value: 6.3
3	OneWire	5	20 min	Register: 1994 Format: 16bit UInt First Byte/Word: 3: LSB/LSW Scale: 2.123	Time: 2016/02/05 09:05:04 Value: 3.2
4	None	1	16 min		---

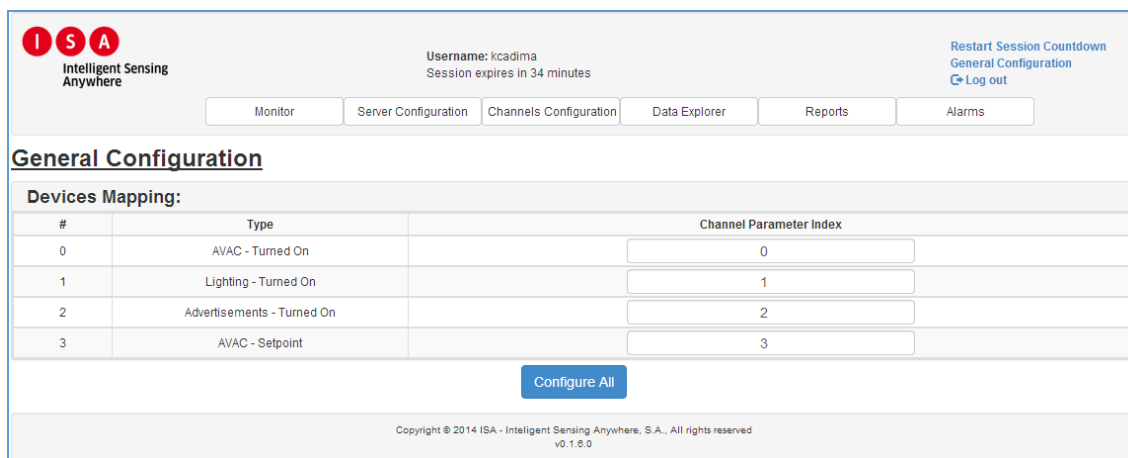
Configure All

Export Channels Configuration to Repository

Figura 50 – Contém um botão no final para exportação de configuração

4.6 General Configuration

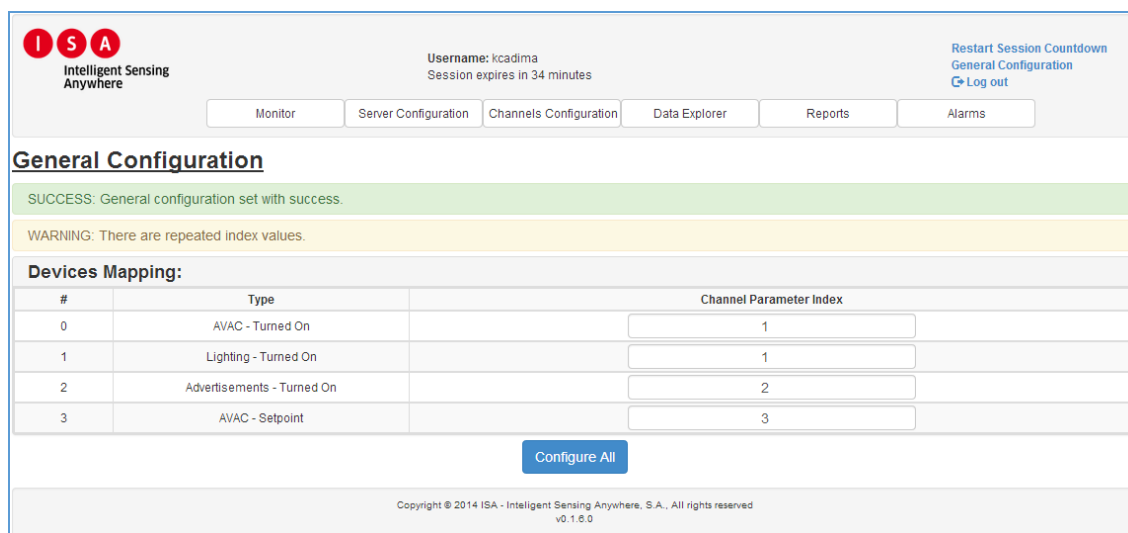
Esta página é dedicada a configurações genéricas. Para já apenas contém a configuração do mapeamento de valores de dispositivos. Escolhe-se o índice do canal de parâmetro para o contexto que lhe é devido.



The screenshot shows the 'General Configuration' page. At the top, there is a navigation bar with the ISA logo and the text 'Intelligent Sensing Anywhere'. The user is logged in as 'kcadima' with a session that expires in 34 minutes. There are links for 'Restart Session Countdown', 'General Configuration', and 'Log out'. Below this is a menu with buttons for 'Monitor', 'Server Configuration', 'Channels Configuration', 'Data Explorer', 'Reports', and 'Alarms'. The main content area is titled 'General Configuration' and contains a section for 'Devices Mapping'. This section has a table with three columns: '#', 'Type', and 'Channel Parameter Index'. The table contains four rows of data. Below the table is a 'Configure All' button. At the bottom of the page, there is a copyright notice: 'Copyright © 2014 ISA - Intelligent Sensing Anywhere, S.A., All rights reserved v0.1.6.0'.

#	Type	Channel Parameter Index
0	AVAC - Turned On	0
1	Lighting - Turned On	1
2	Advertisements - Turned On	2
3	AVAC - Setpoint	3

Figura 51 – Página *General Configuration*, onde é possível a configuração do mapeamento de valores do equipamento.



The screenshot shows the 'General Configuration' page after a successful configuration. At the top, there is a navigation bar with the ISA logo and the text 'Intelligent Sensing Anywhere'. The user is logged in as 'kcadima' with a session that expires in 34 minutes. There are links for 'Restart Session Countdown', 'General Configuration', and 'Log out'. Below this is a menu with buttons for 'Monitor', 'Server Configuration', 'Channels Configuration', 'Data Explorer', 'Reports', and 'Alarms'. The main content area is titled 'General Configuration' and contains a green success message: 'SUCCESS: General configuration set with success.' Below this is a yellow warning message: 'WARNING: There are repeated index values.' Below the warning is a section for 'Devices Mapping'. This section has a table with three columns: '#', 'Type', and 'Channel Parameter Index'. The table contains four rows of data. Below the table is a 'Configure All' button. At the bottom of the page, there is a copyright notice: 'Copyright © 2014 ISA - Intelligent Sensing Anywhere, S.A., All rights reserved v0.1.6.0'.

#	Type	Channel Parameter Index
0	AVAC - Turned On	1
1	Lighting - Turned On	1
2	Advertisements - Turned On	2
3	AVAC - Setpoint	3

Figura 52 – Sucesso na configuração do mapeamento. É mostrado um aviso quando existem valores de índices repetidos no mapeamento.

4.7 Data Explorer/Reports/Alarms

Estas páginas são para ser implementadas e melhor definidas num futuro remoto.

Data Explorer: para visualização de valores sob forma de gráficos.

Reports: para exportar valores para ficheiro segundo uma configuração; obter um documento de revisão e manutenção; exportar os *logs* da interface Web para um repositório remoto.

Alarms: Para configuração de alarmes e respetivas notificações.

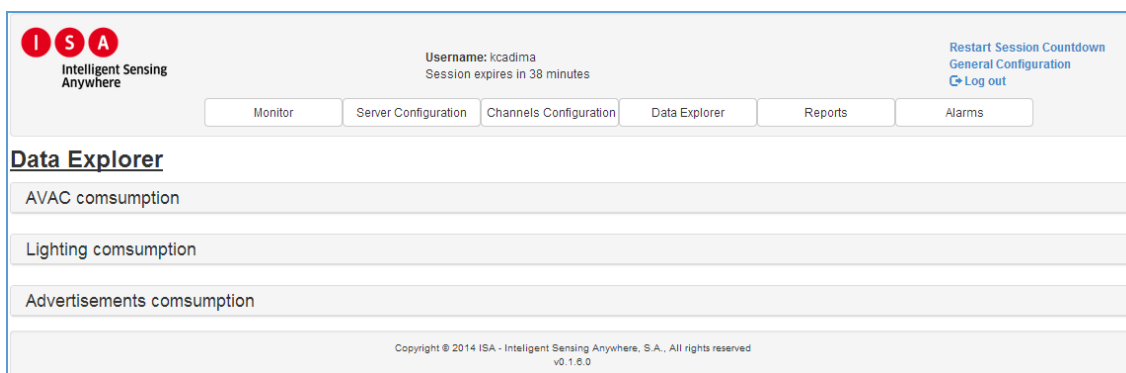


Figura 53 – Página *Data Explorer* no estado atual

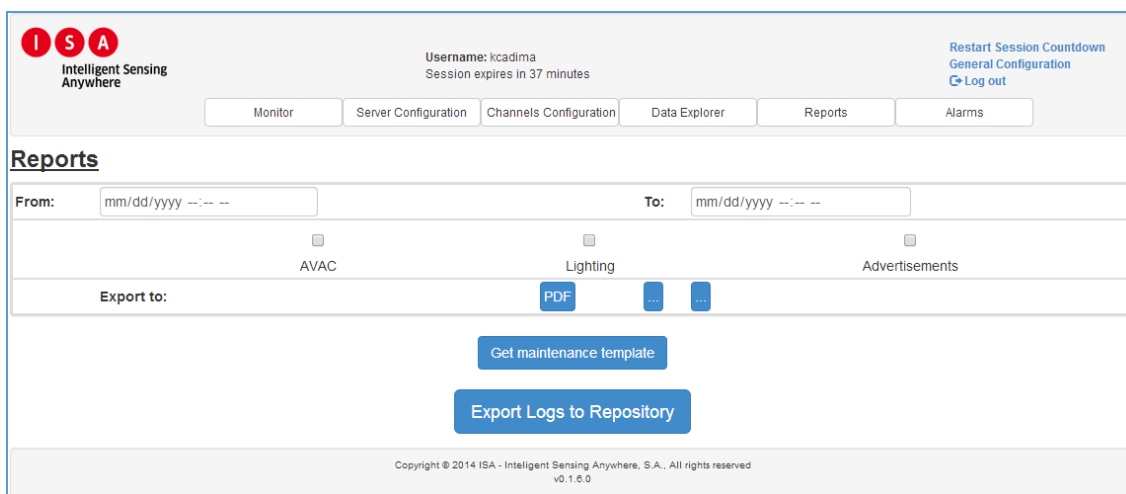



Figura 54 – Página *Reports* no estado atual



**Intelligent Sensing
Anywhere**

Username: kcadima
Session expires in 35 minutes

Restart Session Countdown
General Configuration
[Log out](#)

Monitor Server Configuration Channels Configuration Data Explorer Reports Alarms

Alarms

Notifications: [Update Notification Settings](#)

SMS E-mail

Cell Phone Number:

E-mail Address:

Alarms configuration: [Configure Alarms](#)

	Alarm Description	Priority Level
Current configurations:	AVAC - Not Connected	Low
	Lighting - Not Connected	Average
	Advertisements - Not Connected	High
New configurations:	AVAC - Not Connected	Low <input type="text" value="Low"/>
	Lighting - Not Connected	Average <input type="text" value="Average"/>
	Advertisements - Not Connected	High <input type="text" value="High"/>

Figura 55 - Página Alarms no estado atual

5 Considerações

A interface não está na sua versão final e os protocolos associados não estão finalizados, tendo já definidas alguma páginas da aplicação Web para futura utilização a longo prazo (*data explorer, reports, alarms*).