Mestrado em Engenharia Informática
Estágio
Relatório Final

Desenvolvimento de plataforma para monitorização de consumos em edifícios

Angelo Nicola Simões Iodice
aiodice@student.dei.uc.pt

Orientador DEI:
Professor João P. Vilela

Orientador Streamline:
Engenheiro Paulo Ferreira

Data: 12 de julho de 2012

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

"The best way to predict your future is to create it"
Abraham Lincoln

Agradecimentos

Desde já quero deixar um agradecimento a todos aqueles que fizeram parte do meu percurso académico, bem como aqueles que o cruzaram, ou a minha vida, tendo servido como apoio e motivação no meu crescimento a nível académico, interpessoal e de caráter pessoal.

Agradeço ao meu coordenador de Estágio, Professor João P. Vilela, por servir de mentor na escrita desta Dissertação nesta fase tão importante da minha vida académica e pela sua disponibilização para esclarecimentos de dúvidas.

Deixo aqui também o meu agradecimento ao orientador, por parte da empresa Streamline, Engenheiro Paulo Ferreira, pela disponibilidade imediata e apoio constante no planeamento e desenvolvimento deste projeto. Bem como à empresa pela oportunidade que me foi dada e experiência partilhada, permitindo assim um crescimento pessoal e profissional.

Expresso a minha palavra de agradecimento ao DEEC pela disponibilização de material e de um posto de trabalho fixo, bem como ao DEI pela oportunidade deste Estágio e por me ter acolhido durante longos anos, contribuindo para a minha formação profissional.

A todos os meus amigos presentes e aos ausentes, principalmente pelo apoio e incentivo dado, mas também por acreditarem em mim e me darem forças no dia a dia.

Deixo aqui também a minha palavra de agradecimento ao meu colega de curso, Hugo Vieira, pela sua paciência e ajuda.

Aos meus pais e irmão, conviventes do dia a dia, pelo abrigo e pela oportunidade que me deram de estudar e obter o grau de Licenciado e Mestre em Engenharia Informática numa das mais prestigiadas universidades do país, pela Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

A toda a minha família, portuguesa, italiana e angolana por acreditarem em mim desde sempre, estimulando-me a dar sempre o meu máximo. Dedicando eu este árduo trabalho ao meu avô italiano, por ser a única pessoa que não pode estar presente nesta fase marcante da minha vida.

Por fim, mas não menos importante, expresso as minhas palavras de profundo agradecimento ao meu refúgio, a pessoa devota que me atura há alguns anos, dedicando todo o tempo do mundo ao meu bem-estar, proporcionando-me uma paz e alegria interior imutável, a minha namorada. Tina, estiveste sempre do meu lado, amparando-me incessantemente e motivando-me a ultrapassar os obstáculos que esta vida nos coloca. O meu sincero obrigado, do fundo do coração.

Resumo

Atualmente os edifícios representam uma fatia significativa, não só no consumo de energia, como também na emissão inerente em grandes quantidades de CO2 para a atmosfera. Neste sentido tornou-se fundamental incentivar para um controlo eficaz através da monitorização desses consumos e respetivos custos com o intuito de os reduzir. Surge assim a necessidade de desenvolver software que permite uma recolha de dados energéticos consumidos num edifício de forma a criar, não só práticas de consciencialização no âmbito de uma sensibilização energética educativa, como também relatórios comparativos relacionados com tarifários escolhidos. O trabalho realizado durante o decurso deste estágio, decorre no âmbito do projeto MeWaGo, tendo como área temática central a Eficiência Energética, associada a uma redução de consumos e respetivos custos. Este estágio consiste no desenvolvimento de uma plataforma de monitorização de consumos em edifícios, sendo a integração com os vários dispositivos de recolha de dados energéticos independente do tipo de equipamento. O objetivo fulcral deste projeto baseia-se na ideia de que a plataforma desenvolvida poderá obter registos energéticos em tempo real fornecidos pelos vários equipamentos distribuídos por um edifício, fornecer um conjunto de dados totais e desagregados para análise, gerar relatórios e parametrizar alarmes com notificações, o que permitirá seguir num sentido de eficiência energética.

Palavras-Chave

Consumo Energético, Eficiência Energética, Redução do Consumo Energético, Redução dos Custos Energéticos, Sensibilização Energética, Sistemas de Monitorização de Consumos

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos

API – Application Programming Interface

CO2 – Dióxido de Carbono

CSS – Cascading Style Sheets

CSV – Comma-Separated Values

DC – Diagrama de Carga

DAC – Diagrama Anual de Carga

DEI – Departamento de Engenharia Informática

DEEC – Departamento de Engenharia Eletrotécnica e Computadores

E-R – Entidade-Relação

EDP – Eletricidade de Portugal S.A.

ERSE – Entidade Reguladora dos Serviços Energéticos

FCTUC – Faculdade de Ciências e Tecnologia da Universidade de Coimbra

HTML – HyperText Markup Language

HTTP – Hypertext Transfer Protocol

IDE – Integrated Development Environment

IEEE – Instituto de Engenheiros e Eletricistas e Eletrónicos

iOS – iPhone Operating System

JDBC – Java Database Connectivity

JPEG – Joint Photographic Experts Group

MeWaGo – Measuring of Electricity, Water and Gas & Others

MVC – Model View-Controller

PHP – Hypertext Preprocessor

PME – Pequena e Média Empresa

RCCTE – Regulamento das Características de Comportamento Térmico dos Edifícios

RMI – Remote Method Invocation

RSECE – Regulamento dos Sistemas Energéticos de Climatização em Edifícios

RUP – Rational Unified Process

UML – Unified Modelling Language

TCP/IP – Transmission Control Protocol/Internet Protocol)

SCE – Sistema Nacional de Certificação Energética e da Qualidade do Ar Interior

SDK – Software Development Kit

SDP – Software Design Patterns

XML – Extensible Markup Language

Capítulo 1.
Introdução

O presente documento é referente ao relatório final de estágio realizado pelo aluno Angelo Iodice, no decorrer do ano letivo de 2011/2012, no contexto do projeto MeWaGo. Este estágio foi orientado pelo Professor João Vilela, professor do Departamento de Engenharia Informática da Universidade de Coimbra e pelo Engenheiro Paulo Ferreira, membro da empresa Streamline, Lda.

1.1. Enquadramento

Este estágio surgiu tendo em vista uma colaboração entre o DEI e a empresa Streamline, Lda., experiente na área de Redes e Sistemas de Informação, tendo sido assinado um contrato de prestação de serviços com o DEEC. Este contrato visa a utilização do edifício do DEEC como caso de estudo para o desenvolvimento de um protótipo para uma plataforma de monitorização de consumos em edifícios. Como resultado final proveniente do estágio prevê-se um produto a ser comercializado pela Streamline, tendo como mercado alvo as PMEs e as instituições públicas.

O estágio enquadra-se no contexto da eficiência energética e na sua utilidade crescente atualmente, no âmbito do projeto MeWaGo. Inicialmente este projeto criado em 2007, denominava-se E-Monitor – Sistema de Análise e Monitorização Energética e foi um projeto inovador que permitia proporcionar reduções significativas nos consumos de energia e de água no setor de serviços. Foi um projeto desenvolvido por um grupo de alunos finalistas do curso do DEEC – FCTUC, que inclusive venceu o Concurso Nacional de Ideias Luminosas 2007 – Eficiência Energética[1], promovido pela EDP, introduzido num ambiente de Plano de Promoção da Eficiência no Consumo, da ERSE[2]. Esse grupo era constituído pelos alunos José Silva, Carlos Patrão, Ruben Carvalho e Paulo Ferreira, sob orientação do Professor Humberto Jorge, docente e investigador da FCTUC.

Mais tarde este conceito foi adotado pela empresa Streamline e foi alterado o nome do projeto para MeWaGo. Surgiu a necessidade de criar um sistema único e eficaz, capaz de poder comunicar com qualquer equipamento de contabilização energética já instalado numa edificação, manter um histórico de consumos energéticos através de dados armazenados continuamente, sejam eles totais ou individualizados por localização ou por equipamento de monitorização. É possível assim a utilização deste sistema para fins importantes e diversificados, detalhados de seguida, através da monitorização dos consumos energéticos.

1.2. Motivação

Atualmente, os edifícios representam uma fatia significativa, não só no consumo de energia, como também na emissão inerente, em grandes quantidades, de CO2 para a atmosfera. Em Portugal, existem mais de 3,3 milhões de edifícios que representam cerca de 22% do consumo em energia final. Na última década, o setor dos edifícios de serviços foi o que mais cresceu a nível de gastos energéticos, em cerca de 7.1%, tendo aumentado de 19% para 31% entre os anos 1980 e 1999[3].

Neste sentido têm surgido práticas e políticas de eficiência energética que permitem uma otimização do uso das fontes de energia ao utilizar de forma racional a mesma. A 19 de

outubro de 2006, a Comissão da Comunidade Europeia adotou uma política energética ambiciosa intitulada: "Plano de Ação sobre eficiência energética: Concretizar o Potencial" [4]. Este plano de ação consiste na promoção da eficiência energética, com a meta de redução do consumo de energia final em 20% até 2012[5].

Em Portugal, numa tentativa de concretização dessa meta, foram criadas legislações e regulamentos que promovem o uso dessas práticas em conformidade para cada setor. Das mesmas fazem parte a SCE[6], RSECE[7] e RCCTE[8].

Um dos problemas que serve de entrave ao cumprimento destas legislações, a nível de gestão de energia no setor público, é a falta de informação sobre os consumos energéticos nos processos intrínsecos à atividade em causa. Em muitos destes edifícios encontram-se determinados equipamentos sem que os mesmos tenham contadores individuais instalados, o que impossibilita a determinação dos respetivos registos de consumo, bem como a deteção de situações de consumo anómalo.

Estes obstáculos motivam a criação de sistemas de monitorização próprios para registo de consumos energéticos, entre estes: eletricidade, água e gás. Esses dispositivos tornaram-se úteis por estarem munidos de sistemas a partir dos quais é possível analisar os dados existentes em registo histórico através de uma plataforma própria.

No entanto, cada equipamento tem capacidades diferentes, assim como o seu armazenamento, disponibilização e análise de dados registados são próprios de cada um. É assim necessário criar software compatível com todo o hardware disponível. Neste âmbito surge o desenvolvimento de plataformas de monitorização para consumo de edifícios, dos quais se destaca o MeWaGo.

O MeWaGo é uma solução que permite uma monitorização bastante flexível, adaptando-se facilmente a um edifício, o que permite a interligação a diversos equipamentos já instalados no mesmo, bem como a outros existentes no mercado. O objetivo é manter um registo histórico de consumos energéticos da edificação em causa, assim como uma estimativa dos mesmos, acompanhada de um registo de custos inerentes. Assim será possível englobar as funcionalidades essenciais de deteção de consumos anómalos, o que contribuirá para uma redução significativa do consumo e custos associados. É importante também induzir os utentes de uma edificação a tomar atitudes comportamentais e de consciencialização que levem a uma utilização eficiente de energia, facultando informação sobre o impacto ambiental devido ao consumo energético do mesmo.

1.3. Objetivos

No contexto do projeto E-Monitor já se encontrava criada uma primeira versão de um protótipo básico e experimental para esta plataforma numa versão web, pela equipa envolvida inicialmente. A mesma tinha criado um modelo de dados básico que se encontrava populado com alguns valores para efeitos de teste. Esta equipa estruturou o sistema por quatro camadas distintas, sendo que todo o hardware necessário foi instalado para efeitos de estudo e testes no DEEC num projeto posterior. A instalação destes equipamentos de aquisição e transmissão de dados energéticos reforçou a ideia de que a estrutura física aplicada pode ser qualquer uma, desde que a plataforma a desenvolver consiga tirar partido das funcionalidades de maior importância num projeto desta envergadura. A Figura 1 representa as camadas existentes no conceito do projeto anterior.

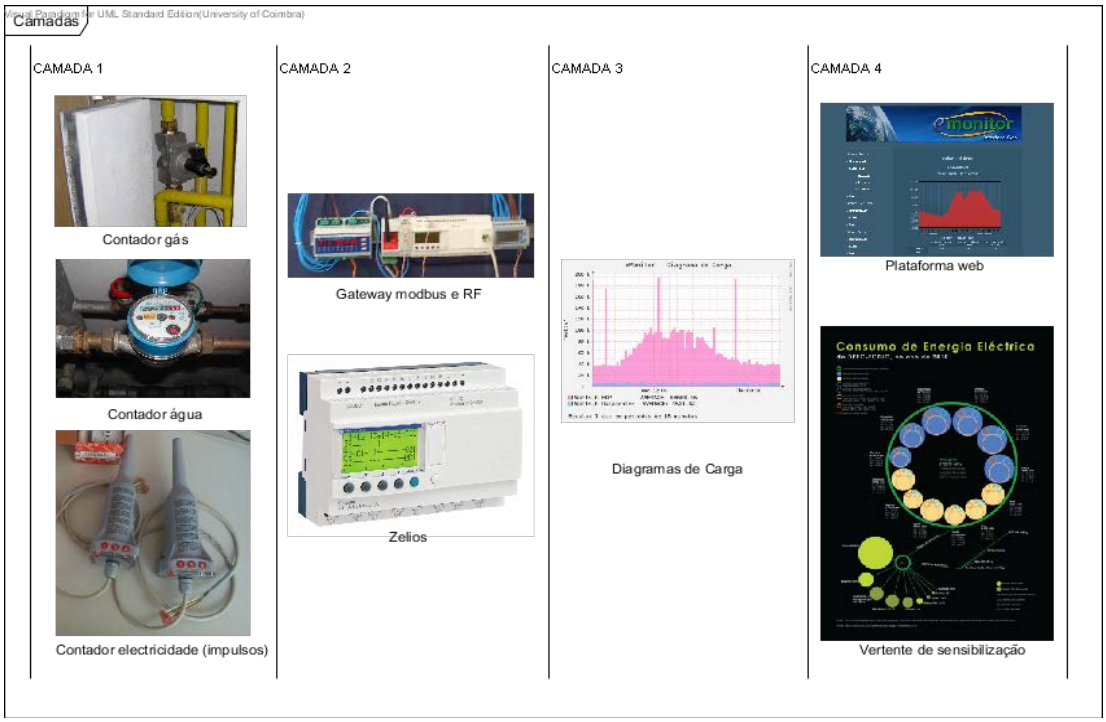

Figura 1 – Camadas do projeto MeWaGo

A Camada 1 representa a parte física constituída pelos equipamentos de aquisição de leituras de registos que serão reencaminhadas para a camada seguinte. Na Camada 2 essas leituras de dados são concentradas, analisadas e armazenadas, no qual entram a leitura dos protocolos Modbus (e.g. TCP, RTU e ASCII)[9] e ligações em série RS (e.g. RS-232 e RS-485)[10] que não fazem parte de discussão neste relatório. A Camada 3 representa a análise e armazenamento desses dados na base de dados de um servidor central, retirados do DC de cada equipamento. Por fim, na Camada 4 existe a apresentação da informação e interação com o utilizador através de várias aplicações cliente.

Este sistema estava inicialmente agregado apenas ao hardware presente no edifício no qual se estava a testar este projeto, o que tornou necessário criar um modelo mais geral e abrangente. Este terá que ser o mais abstrato possível para se poder integrar mais facilmente noutros edifícios e noutros cenários.

Uma vez que o projeto antigo era um pouco rudimentar, incompleto e apenas podia ser instalado no DEEC, tornava-se difícil a sua integração noutros edifícios. Tornou-se portanto necessário redesenhar todo o software pertencente às Camadas 3 e 4 da figura anterior. Esta tarefa permite uma nova abordagem mais flexível no que diz respeito à implementação em edifícios com características diferentes. Desta forma, surge a necessidade de enunciar os objetivos principais na construção do *core engine* (*backend*) da Camada 3, cujo produto final será uma versão Release Candidate:

- Integração flexível de equipamento diversificado de leitura de grandezas energéticas;
- Tratamento e armazenamento padrão de informação recolhida para processamento futuro.

A utilização desta informação servirá para um leque variado de aplicações da Camada 4 que irá permitir efetuar análises energéticas, económicas e ambientais de um ou mais edifícios. A equipa E-Monitor conceptualizou previamente esta plataforma (*frontend*) em três interfaces distintas:

- Gestão Energética;

- Gestão Financeira;
- Sensibilização Ambiental.

A primeira permite ao utilizador efetuar uma análise técnica da informação recolhida, como por exemplo, gráficos de consumo, análise evolutiva e previsões de consumo. É possível também comparar séries de dados guardados em histórico de intervalos de tempo à escolha.

A segunda efetua uma análise financeira dos dados obtidos, através da utilização de tarifários, previsões de custos e localização de oportunidades de racionalização de consumo.

A terceira, focada no utente *freerider* do edifício, consiste numa vertente de sensibilização/consciencialização para o consumo do edifício e o impacto ambiental que este produz, direta e indiretamente, com o objetivo de promover práticas de eficiência energética.

Esta última componente não irá ser considerada no âmbito deste estágio, pelo que será apenas necessário criar um pequeno protótipo de teste para demonstração de algumas das funcionalidades importantes do *backend* implementado. Para tal foi decidido implementar uma aplicação para desktop em java, resultando o seu produto final numa versão Beta.

Surgiu também a necessidade de criar uma aplicação de demonstração para dispositivos iOS (iPhone). O objetivo do desenvolvimento desta aplicação, com funcionalidades básicas de visualização de consumos, reflete-se no fato de estudar a potencialidade destes conceitos em dispositivos móveis. Este pequeno protótipo será entregue como uma versão Alfa.

1.4. Estrutura do Documento

Este relatório de estágio está estruturado em 6 capítulos.

O Capítulo 1, onde se insere esta descrição, consiste num enquadramento deste documento no projeto MeWaGo, no qual são descritas as propostas de valor do mesmo, seu âmbito e objetivos primordiais.

O Capítulo 2 abrange uma área vasta de investigação sobre o Estado da Arte, no qual se encontra descrito um conhecimento do campo em investigação. Desta faz parte um estudo de ferramentas utilizadas no mesmo âmbito de investigação e uma análise sobre o contributo inovador deste projeto. É descrita toda a informação relacionada com ferramentas comercializadas já existentes na área, da qual faz também parte uma breve conclusão comparativa entre a concorrência e o MeWaGo.

O Capítulo 3 clarifica os objetivos do projeto de estágio, enuncia com detalhe o planeamento, abordagem e metodologia seguidos no desenvolvimento do software.

O Capítulo 4 dá uma ideia global do trabalho executado e do progresso realizado no sentido de cumprir com os objetivos do projeto. Deste faz parte a análise de requisitos dos diferentes módulos implementados e a definição da arquitetura da plataforma.

O Capítulo 5 descreve sucintamente todo o progresso de trabalho de estágio, desde o estudo do conceito da plataforma, passando pelo desenvolvimento de um pequeno simulador de tráfego que possibilita o carregamento de dados, até aos produtos finais desenvolvidos.

O Capítulo 6 contém algumas conclusões retiradas, recapitula a abordagem ao projeto de estágio corrente de modo a clarificar a relevância científica do mesmo. Existe também referência ao trabalho que no futuro possa trazer uma mais valia a todo o trabalho feito.

Capítulo 2.
Estado da Arte

Neste capítulo é descrito detalhadamente todo o estudo feito sobre o Estado da Arte da área temática deste projeto de estágio. É aqui feita uma análise detalhada de sistemas de monitorização existentes atualmente no mercado e seu modo de funcionamento, seguindo uma breve conclusão sobre a comparação das vantagens e desvantagens destes com a plataforma MeWaGo. Mais do que isso, são também analisadas soluções para dispositivos móveis e bibliotecas para geração de gráficos, uma vez que esta área é um ponte forte a explorar neste projeto.

2.1. Sistemas de Monitorização

Os consumos energéticos e custos associados em edifícios de grande envergadura são, por regra, demasiado elevados. Torna-se portanto necessário efetuar um controlo eficaz e por sua vez tomar ações imediatas de mitigação do mesmos. Para tal, têm sido desenvolvidos sistemas de monitorização capazes de tratar dados recolhidos de dispositivos energéticos existentes nessas edificações, capazes de fornecer uma análise detalhada de consumos, ajuste de tarifas de custos agregados e deteção de valores anómalos em tempo real. A integração destes sistemas no mercado veio aprimorar esse tipo de tarefas, antes irrealizável, o que permitiu atenuar o tempo de intervenção, quando necessário, no âmbito da eficiência energética e redução de custos inerentes aos processos e serviços fornecidos pelos edifícios.

Neste sentido surgiram diversas empresas que começaram a desenvolver software e hardware específico para sistemas de automação onde são implementadas algumas soluções de monitorização que nem sempre satisfazem as necessidades dos gestores técnicos e financeiros dos edifícios. Os mesmos são constituídos por material computacional que recolhe dados de equipamentos implementados em soluções proprietárias dos fabricantes. A evolução nesta área tem ocorrido principalmente ao nível do armazenamento, processamento e modo de visualização intuitivo de valores energéticos, na qual pode ser gerido o registo histórico de informação via web ou através de *displays* localizados em pontos-chaves no edifício em que é instalado[11] [45].

Estes sistemas e correspondentes medições de valores energéticos são essenciais para a implementação de estratégias de otimização energética e apoio nas auditorias energéticas no âmbito do SCE[6].

2.2. Análise de Plataformas de Monitorização

Para além dos sistemas de gestão energética existentes para ambientes domésticos, existem no mercado para o setor não residencial (serviços, indústrias, hotelaria, entre outros) outras soluções e produtos próprios. Foram escolhidos para esta fase os dois tipos de produtos que melhor se identificavam com as funcionalidades básicas de monitorização de consumos energéticos pretendidas para o MeWaGo.

2.2.1. Lista de Produtos Concorrentes

A Tabela 1 mostra a existência de funcionalidades gerais e diferenciadoras nas plataformas que mais se destacam nesta área, podendo equipará-las com as da plataforma MeWaGo.

Parâmetros/Produtos	Serious Energy[12]	Open Energy Monitor[13]	Opower[14]	Silver Sprint SEP[15]	eMonitor[16]	Tendril Connect[17]	Wi-LEM[18]	Energy Hub[19]	Si-Master[20]	Energy Brain[21]	Optimal Monitoring System[22]	eSight[23]	MeWaGo
Funcionalidades de Automação				X	X	X		X					
Integração apenas com equipamento próprio		X	X	X	X	X	X	X	X	X	X		
Gestão centralizada de edifícios	X		X	X			X				X		X
Monitorização de múltiplos pontos de leitura	X	X	X	X	X		X			X	X	X	X
Notificação de alarmes personalizados			X	X	X			X	X	X	X	X	X
Interfaces virtuais										X			X
Análise detalhada de consumos e custos		X	X	X	X	X		X	X		X	X	X
Adaptação de diversas aplicações cliente		X			X	X		X					X
Comparação com tarifários	X			X					X	X	X	X	X
Previsões e estimativas	X								X			X	X

Tabela 1 – Tabela comparativa de sistemas de monitorização

Todos os produtos pesquisados tiveram como base as funcionalidades de análises diárias, semanais, mensais e anuais de consumos e/ou custos de grandezas tais como a eletricidade, gás e água.

Foram analisados muitos mais produtos dos propostos na tabela anterior, no entanto por possuirem um pequeno conjunto de funcionalidades que não permitia uma comparação correta entre todas as soluções, foram postos de parte. Outro fator deveu-se à especificidade a certas áreas, não sendo tão abrangentes, como por exemplo, serem focados só em automação ou apenas na análise de consumos, custos ou pégada ecológica. Produtos que continham menos de quatro funcionalidades das expostas na tabela anterior não foram indicadores corretos de comparação, nem produtos dos quais não existia informação detalhada sobre as suas funcionalidades para efeitos de análise ou produtos descontinuados.

2.2.2. Conclusões

A título conclusivo, os produtos apresentados possuem um leque de características em comum. No entanto, alguns são focados na análise de consumos energéticos e respetivos custos, enquanto que outros apenas se limitam a uma recolha de dados de contadores para uma posterior análise energética rudimentar.

De entre os vários produtos estudados, o MeWaGo será uma das únicas plataformas portuguesas que pretende ser independente do equipamento de monitorização utilizado, sendo necessário existir uma integração de novos dispositivos mais fácil. A análise energética de muitos dos produtos anteriormente mencionados é superficial, pois estes têm como objetivo principal a gestão de sistemas de automação. O MeWaGo afasta-se completamente da automação, aspirando a ser um produto na gama da eficiência energética de baixo custo, aplicado a grandes complexos. Esta será uma solução mais equilibrada que junta o melhor de todos os produtos vistos até agora.

As funcionalidades abrangidas, passam desde a monitorização de energia elétrica em tempo real (consumos de energia e potência respetiva de gás, água, temperatura, humidade e muitos outros), até à microgeração (input/output), criação de relatórios técnicos e financeiros, desagregação de consumos e custos através de diversos parâmetros, consequente auxílio na obtenção da melhor opção tarifária da instalação e deteção de fugas e consumos anómalos com acionamento de alarmes e envio de notificações. Para além do conjunto ilimitado de pontos a monitorizar, será possível a criação de pontos virtuais de leitura, constituídos por diferentes operações entre pontos reais de leitura à escolha. Será também possível utilizar este fator conforme a gestão centralizada de edifícios patente no MeWaGo, criando assim pontos próprios de medição.

Graças à importância dos dispositivos móveis hoje em dia, misturado com a relevância temática da Eficiência Energética, pretende-se aprofundar estes conceitos em ambiente móvel, para iOS, o qual será alvo de análise cuidada por parte do MeWaGo.

2.3. Análise de Soluções

Uma vez que da secção anterior fez parte uma análise detalhada de produtos na área de eficiência energética, tendo sido comparada a plataforma numa só (*backend* + *frontend*), esta secção será mais dirigida ao estudo de soluções para iPhone e bibliotecas para geração de gráficos que foram examinadas para poder dar resposta às funcionalidades de maior valor da plataforma MeWaGo (gráficos de consumos e custos).

2.3.1. Aplicações iPhone

Com a importância que aplicações para dispositivos móveis têm no mercado atual, foi escolhida a plataforma iOS da Apple para desenvolver um protótipo para demonstração de algumas funcionalidades básicas da plataforma MeWaGo, permitindo a utilização nos três seguintes dispositivos: iPhone, iPod Touch e iPad. Apesar de ainda não existirem aplicações especificamente relacionadas com consumos e custos associados a grandes edifícios, irão ser aqui analisadas brevemente algumas das mais importantes aplicações de gestão de consumos existentes para ambientes domésticos. Das aplicações encontradas, foram exploradas apenas as seguintes três que mais se inclinam para o tipo de utilização final do MeWaGo iOS:

MeterRead[24]
Criada pela Zerogate[25], permite ao utilizador visualizar dados de eletricidade para monitorizar e gerir consumos de energia de uma casa. Apesar de ser feita uma previsão mensal de consumos, o utilizador é que terá que inserir o primeiros dados, o que tira um pouco a parte automática do mecanismo principal. Possui apenas estas funcionalidades.

Energy UFO[26]
Desenvolvida pela Visible Energy Inc.[27], esta ferramenta dá feedback ao utilizador sobre consumos e custos de sua casa através de um interface bastante apelativo. No entanto, o facto de apenas existir uma versão beta com simulação de valores e apenas existir medição de eletricidade proveniente de dispositivos da empresa, limita um pouco a integração com outros dispositivos, o que deixa um pouco a desejar.

Meter Readings[28]
Esta aplicação, de Graham Haley[29], foi a que mais se distinguiu na pesquisa. Permite a visualização de consumos e custos diários, semanais e mensais, de eletricidade, água e gás. Até permite integração com iCloud, Dropbox, email, Wi-Fi, Bluetooth e importação de ficheiros CSV para backup (após feito um upgrade). A interface é bastante apelativa, no entanto esta ferramenta cinge-se apenas a tarifários locais (UK) e é aplicada apenas a ambientes domésticos.

2.3.2. Bibliotecas para geração de gráficos

Uma vez que terá que ser fundamental demonstrar o funcionamento da plataforma MeWaGo implementada, é imprescindível a utilização de bibliotecas para geração de gráficos. São de seguida alvo de estudo, todo o tipo de bibliotecas relacionadas com as aplicações de demonstração a desenvolver, sendo elas bibliotecas para geração de gráficos em java para a aplicação MeWaGo Desktop e para dispositivos móveis para o MeWaGo iOS. Em alternativa, foi estudada a possibilidade de utilização de gráficos para versões web e possível integração com web browsers das aplicações a desenvolver. Deu-se prioridade às soluções mais conhecidas em voga e em versões mais atuais.

2.3.2.1. Java

jCharts[30]
Esta biblioteca open-source é 100% baseada em java e fornece uma variedade de gráficos. Pode ser integrava com Servlets, JSP´s e aplicações Swing. A documentação existe em abundância e detalhada, no entanto possui gráficos muito rudimentares, interface pouco apelativa e não permite criar alguns gráficos que serão necessários para o MeWaGo.

EasyCharts[31]

A Object Planet Inc[32], desenvolveu esta biblioteca munida de bastante documentação facilmente percetível, sendo a mesma reconhecida pela sua simplicidade de utilização. Contudo, para além de alguns gráficos serem pouco intuitivos, é necessário obter licenças bastante caras para utilização destes gráficos.

JFreeChart[33]
Esta ferramenta open-source, distribuída por David Gilbert e Thomas Morgner, é um pouco complexa para integração e a documentação é escassa. Um outro problema é o facto de que o carregamento de dados pode ser pesado e tornar a geração de gráficos bastante lenta. No entanto possui uma variedade de gráficos úteis para integração com o MeWaGo.

SW2D[34]
A empresa JenSoft, disponibiliza esta ferramenta open-source para utilização em ambiente java (aplicações Swing desktop, Applet ou Servlets). É uma solução bastante completa, próxima de uma aplicável ao projeto MeWaGo pela variedade e apresentação dos gráficos disponibilizados. No entanto a sua aplicação pode-se tornar um pouco complexa.

2.3.2.2. iOS

iOS:Chart[35]
Esta biblioteca tem uma variedade de gráficos bastante interessante e a sua interface é bastante apelativa, focando as potencialidades dos gráficos 3D. Existe documentação própria para a API e parece de fácil integração. No entanto as suas licenças são bastante caras.

iPhone Charting Library [36]
Apesar de existir documentação cuidada para esta biblioteca, não existem visualizações para verificar a potencialidade desta ferramenta. A codificação é bastante simples, no entanto, e mais uma vez, a existência de licenças torna a utilização desta biblioteca não viável.

Core Plot[37]
Os gráficos desta biblioteca, à primeira vista parecem bastante fluidos. A documentação é exaustiva e a sua integração no iPhone parece difícil, apesar de possuir uma licença BSD a seu favor.

2.3.2.3. Web

Google Chart Tools[38]
A Google tem a sua ferramenta própria de geração de gráficos. Possui documentação organizada e bons exemplos. Não necessita de licença, no entanto esta biblioteca depende da plataforma Google, pois é necessário recorrer a folhas de excel do Google docs para guardar dados o que iria complicar o seu uso na plataforma MeWaGo.

Fusion Charts[39]
Esta biblioteca permite a geração de gráficos em ambiente flash, para tal tem um contra, ser necessário a instalação do adobe flash player, o que poderá tornar a geração de gráficos um pouco pesada. De resto, a variedade de gráficos intuitivos e a existência de documentação é imensa, apesar da sua integração e utilização ser mais complexa.

amCharts[40]
Esta biblioteca open-source destacou-se pela variedade de gráficos, a sua interação e visualização intuitiva. É possível efetuar uma personalização dos seus gráficos, para além da

documentação ser vasta, de fácil leitura e aplicação e o feedback e contacto é imediato graças à comunidade existente. Esta ferramenta não necessita de licenças para o efeito desejado e encontra-se em constantes atualizações.

2.3.3. Conclusões

Em relação às aplicações iOS encontradas, verificou-se que não existe ainda um grande leque de soluções adaptadas às necessidades presentes em edifícios. Para além de não ter sido encontrado nenhum produto português nesta área, todas as soluções encontradas carecem de algumas funcionalidades das quais a futura aplicação MeWaGo iOS irá benefeciar. As mesmas tendem a monitorizar consumos em ambientes domésticos, dificultando uma futura integração noutros cenários, em que a monitorização de diversos edifícios ou diversos pontos é importante. Desta forma, o MeWaGo iOS poderá começar a singrar neste mercado como um produto inteiramente português, que permita responder a diversas necessidades importantes às quais as soluções encontradas não permitem ainda responder.

Das bibliotecas de geração de gráficos pesquisadas, chegou-se à conclusão de que se deveria adotar uma solução web que pudesse ser integrada com web browsers nas mais diversas aplicações cliente. Para além de motivos relacionados com exclusão de produtos que necessitassem de licenças, que não era do interesse da empresa, foi escolhido a biblioteca open-source amCharts por se destacar nesta gama de geração de diversos tipos diferentes de gráficos interativos. Assim, será possível utilizar a integração de web browsers nativos de dispositivos móveis, ou utilizando bibliotecas Swing java, para permitir o acesso a páginas que geram gráficos, podendo assim uniformizar este processo. Esta solução web foi também escolhida tendo em vista melhorias da performance na geração de gráficos do lado do servidor e não no lado do cliente.

Capítulo 3.
Objetivos, Planeamento e Metodologia

Neste capítulo abordo a gestão do projeto utilizada no curso deste projeto. As seguintes secções pretendem descrever o planeamento idealizado, a organização da equipa e a metodologia seguida para atingir os objetivos propostos deste estágio.

3.1. Clarificação dos Objetivos

Por forma a dar continuidade ao decorrer deste projeto utilizei a definição em camadas do projeto anterior e aproveitei o levantamento das necessidades já efetuadas pelo aluno Francisco Ferreira no âmbito da sua dissertação relacionada com o projeto E-Monitor[45] [46]. Utilizei informação deste documento para analisar padrões de consumos, o que permitiu a construção de um modelo de dados completo e o desenvolvimento de um simulador de tráfego. Deste modo, os objetivos específicos deste estágio que trarão algum benefício ao projeto atual são os seguintes:

- Elaboração de um documento técnico de Especificação de Requisitos de Software;
- Modelação de uma nova base de dados;
- Elaboração de um documento de Arquitetura de Software;
- Desenvolvimento de um pequeno simulador de teste de tráfego para simulação e *dumping* de dados;
- Design de *mockups* relativos ao *frontend* (versão web, desktop e mobile);
- Desenvolvimento do *backend* (servidor, proxy e drivers);
- Desenvolvimento de protótipos de demonstração para o *frontend* (aplicação desktop, aplicação iOS e dashboard do DEEC).

A elaboração do documento de Especificação de Requisitos de Software englobou as necessidades levantadas desde o começo do projeto até à sua fase atual, no qual adicionei mais requisitos funcionais que achei relevantes e um novo módulo para Administração do software. Criei um novo modelo de dados para mitigar as falhas a que o primeiro não se encontrava preparado. Esta tarefa permitiu o desenvolvimento do simulador de tráfego que irá ser fruto de análise no Capítulo 5. Elaborei um documento de Arquitetura de Software, uma vez que não existia nenhum documento com a definição da arquitetura a utilizar neste projeto.

Em relação à aplicação iOS, o intuito da mesma é o controlo eficaz de consumos num ou mais edifícios. Para tal, o plano desta componente foi o seguinte:

- Estudo de ferramentas de desenvolvimento para iOS em ambiente Mac;
- Design de *mockups* de interface para a aplicação;
- Desenvolvimento da aplicação;
- Elaboração de manual de utilizador e documento de testes.

3.2. Planeamento

Com vista ao cumprimento dos objetivos acima clarificados, foi delineado o planeamento de trabalho, representado na Figura 2 – Diagrama de Gantt do planeamento de estágio, através da elaboração de um Diagrama de Gantt no GoogleDocs. Esta figura ilustra o trabalho levado a cabo durante o período do primeiro semestre e o trabalho realizado durante o segundo semestre.

Figura 2 – Diagrama de Gantt do planeamento de estágio

A figura anterior representa o trabalho de estágio decomposto em seis fases fundamentais e distintas, durante o período de 1 de setembro de 2011 até 29 de junho de 2012.

Na primeira fase realizou-se um estudo detalhado sobre o Estado da Arte. Efetuei um estudo sobre o conceito da plataforma e analisei documentação. Uma vez que este projeto se encontra em fase de maturação e é necessário redesenhar o sistema MeWaGo, optou-se por verificar produtos já existentes nesta área de foco para detetar assim novos requisitos para a implementação futura. Para esta fase foram reservadas apenas duas semanas.

A segunda fase, de maior importância, correspondeu a uma Análise de Requisitos e serviu para elaborar um documento com as especificações técnicas para o desenvolvimento do software. Criei este documento com o maior detalhe possível para servir de suporte para o decorrer da fase de implementação. Para esta fase foi alocado um mês e meio.

A terceira fase serviu para elaborar todo o Design da Arquitetura da plataforma. Comecei por analisar padrões de consumos das grandezas envolvidas utilizando dados recolhidos no DEEC. Este estudo permitiu ter uma noção concreta dos dados a processar. Defini um novo diagrama do modelo da base de dados a utilizar, onde se pretendeu verificar se conseguia atingir um nível de abrangência maior que o modelo de dados antigo. Começou-se a pensar sobre a arquitetura do software a desenvolver na qual elaborei um documento de arquitetura de software, bem como *mockups* alusivos à plataforma web, aplicação desktop e iOS. Nesta etapa implementei também um simulador de tráfego para carregamento de dados na base de dados que idealizei. Este simulador permite a inclusão de novos dispositivos de uma forma bastante acessível. Para esta fase foram reservados dois meses.

Na quarta fase deu-se início à fase de Implementação, da qual fazem parte: o *backend* e os protótipos para o *frontend* (aplicação desktop, aplicação iOS e dashboard do DEEC). Esta

fase, de maior duração, teve como base as especificações elaboradas na segunda fase e a arquitetura delineada na terceira fase. Para esta fase foram alocados 5 meses e meio.

Os últimos dias foram reservados para a quinta fase, referente a testes finais e avaliação de resultados para o *backend* e *frontend*.

A última e sexta fase deste plano de estágio, já fora do plano de estágio, consistiu na produção do relatório final que detalha toda a atividade que foi realizada durante o estágio.

É de referir que não existiu qualquer tipo de atraso no cumprimento de todas as tarefas primárias e secundárias durante o primeiro semestre. Já no segundo semestre, o primeiro mês (janeiro) foi mais dedicado a algumas alterações da arquitetura projetada inicialmente.

3.3. Equipa

A Equipa foi composta por três pessoas envolvidas no processo de desenvolvimento do MeWaGo. Para além de mim, as outras duas pessoas envolvidas que fizeram parte dele e seus papéis foram as/os seguintes:

- Engenheiro Francisco Maia:
    - Esclarecimento de dúvidas sobre os requistos;
    - Aprovação de documentos elaborados;
    - Validação do trabalho realizado.
- Engenheiro Paulo Ferreira:
    - Orientação do estagiário, esclarecimento de dúvidas;
    - Esclarecimento de dúvidas sobre os requisitos;
    - Aprovação de documentos elaborados;
    - Validação do trabalho realizado.

O orientador de estágio na empresa Streamline, Engenheiro Paulo Ferreira, assumiu uma função marcante no controlo de qualidade de documentos e sua aprovação, bem como o Engenheiro Francisco Maia. Tal aconteceu, uma vez que eram marcadas reuniões regularmente para verificar o estado do trabalho no âmbito deste estágio.

O Engenheiro Francisco Maia, pertencente à área de Informática, foi o foco principal de ajuda no decorrer da fase de implementação do sistema, pois foi o que melhor me pode esclarecer a nível técnico, enquanto que o segundo pôde esclarecer dúvidas mais teóricas.

O Engenheiro Paulo Ferreira, orientador deste projeto, pertence à área de Eletrotécnica e é a pessoa com fortes conhecimentos sobre o projeto que ajudam na clarificação das necessidades e requisitos do sistema.

No entanto, estes dois elementos não participaram ativamente e todo o trabalho envolvido neste projeto esteve a cargo do estagiário, servindo os mesmos apenas para suporte, mas permitindo uma oportunidade de exploração e investigação autónoma e autodidata.

3.4. Metodologia

Para o desenvolvimento do MeWaGo, durante o segundo semestre foi utilizada uma metodologia ágil de desenvolvimento de software iterativa e incremental – SCRUM.

Esta metodologia permite exercer maior controlo sobre o processo de desenvolvimento do produto, devido às várias iterações inerentes e constante *feedback* do cliente. Além disso, permite detetar desde cedo, desvios no Roadmap/requisitos do produto pela mesma razão.

A Figura 3 – Processo SCRUM (imagem retirada de [41]) representa o processo inerente a esta metodologia.

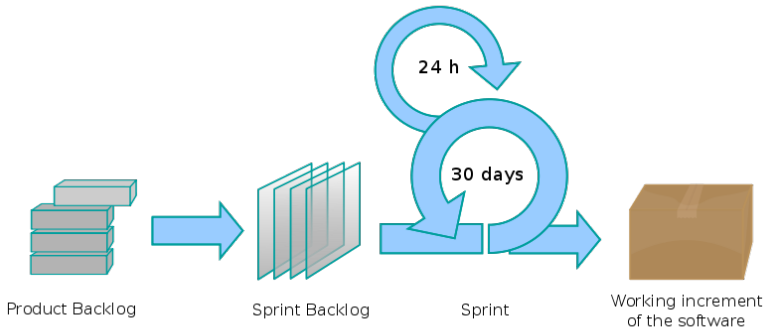

Figura 3 – Processo SCRUM (imagem retirada de [41])

Existem três tipos de intervientes em todo o processo. Desde o Product Owner (Francisco Maia), responsável por manter o Product Backlog, representando os interesses dos clientes e assegurando o valor do trabalho desenvolvido pela Equipa de Desenvolvimento, até ao Scrum Master (Paulo Ferreira) que foi a pessoa responsável pela aplicação correta de todo processo de modo a maximizar os seus benefícios. Ambos fizeram parte da Equipa de Desenvolvimento em alguns módulos que irão ser especificados no Capítulo 5. Esta Equipa trata de desenvolver código e entregar incrementos do produto, no final de cada iteração de desenvolvimento. A estas iterações dá-se o nome de Sprints.[42]

À partida, o Product Owner lista as funcionalidades de alto nível do produto a desenvolver por ordem de prioridade, criando assim o Product Backlog. Existem várias iterações para implementação de todas estas funcionalidades, denominadas Sprint. O objetivo destas é a implementação de funcionalidades escolhidas pelo Scrum Master numa reunião inicial de Sprint, que poderá durar alguns dias. No final de cada Sprint é feito uma reunião de retrospetiva que permite analisar o que correu bem e o que correu mal. Existem também reuniões diárias (Daily Scrum) que permitem ao Scrum Master acompanhar a performance da Equipa de Desenvolvimento e auxiliá-la focando-se em três perguntas importantes: "o que foi feito?", "o que irá ser feito?" e "que dificuldades existem?".

Para este projeto, o Scrum Master decidiu criar Sprints de sete dias e todos os artefactos relacionados com esta metodologia (Product Backlog e Sprints Backlogs) poderão ser consultados com maior detalhe no documento de Planeamento [Anexo E].

Uma vez que foi necessário trabalhar em equipa, foi utilizado o Turtoise SVN[43] como repositório Subversion e o Bugzilla[44] para controlo de tarefas, bugs/erros e tempo estimado de desenvolvimento das funcionalidades presentes num determinado Sprint.

Capítulo 4.
Arquitetura e Especificação de Requisitos

Neste capítulo abordo todos os aspetos relativos, não só ao trabalho efetuado no decorrer do primeiro semestre de estágio, como também ao progresso realizado no sentido de preparação para a fase de implementação decorrida durante o segundo semestre. As seguintes secções deste capítulo descrevem todo o trabalho efetuado em relação ao levantamento de requisitos de ambas as componentes e definição da sua arquitetura geral.

4.1. Análise de Requisitos

Foram encontradas e definidas duas grandes componentes na plataforma MeWaGo. São elas: o *backend* e o *frontend*. O primeiro é composto por um Servidor e por múltiplas instâncias com alguns módulos idênticos e outros mais personalizados, denominadas Proxy. Desta fizeram também parte um conjunto de drivers de acesso a dispositivos de leituras de grandezas energéticas. O segundo é composto por duas aplicações cliente, sendo de destaque neste estágio uma aplicação desktop e uma para iOS.

Segundo a metodologia adotada para este projeto foram sendo realizadas reuniões internas que permitiram fazer um levantamento de requisitos cuidado e sua escrita formalizada que ajudaram na definição do âmbito do projeto e respetivos casos de teste. Estes processo foi importante para a geração de documentação e artefactos importantes para a criação do Product Backlog.

O resultado final com maior detalhe desta fase pode ser consultado no Capítulo 3 do documento de Especificação de Requisitos de Software [Anexo A]. Este documento foi elaborado utilizando o template da IEEE, disponibilizado para este tipo de documentos[47] [48].

4.1.1. Backend

Nesta secção pretende-se descrever a análise de requisitos realizada para o *backend*. Uma vez que esta componente sempre foi uma prioridade no plano de estágio, um pouco por ser necessário redesenhá-la, são aqui descritas as novas funcionalidades necessárias para o Servidor, Proxy e Drivers. São de seguida apresentados os requisitos funcionais e não funcionais de alto nível para as duas componentes do *backend*.

4.1.1.1. Servidor

Requisitos funcionais

- Funcionalidades Core:
    - Mecanismo de sincronização;
    - Temporizador para despoletar recolha de dados de todos os Proxys de 15 em 15 minutos;
    - Mecanismo de notificação de alarmes (email e iPhone);
    - Mecanismo de fail-over (salvaguarda num sistema de ficheiros local de conecções de Proxys para utilização futura quando reiniciar o serviço);

  - Tratamento de exceções.

- Funcionalidades de conecção:
  - Pedir valores das leituras ao Proxy;
  - Reiniciar configuração do Proxy;
  - Enviar valores de configuração para Proxy;
  - Enviar valores de funcionalidades de autenticação;
  - Enviar valores de funcionalidades administrativas;
  - Enviar valores de funcionalidades de gestão técnica;
  - Enviar valores de funcionalidades de gestão financeira;
  - Enviar valores de funcionalidades de sensibilização.

- Funcionalidades de gestão:
  - Recolher valores de funcionalidades de autenticação ;
  - Recolher valores de funcionalidades administrativas;
  - Recolher valores de funcionalidades de gestão técnica;
  - Recolher valores de funcionalidades de gestão financeira;
  - Recolher dados de funcionalidades de sensibilização.

- Funcionalidades de manipulação de dados da base de dados:
  - Guardar dados de valores de leitura dos Proxys;
  - Apagar dados de valores de leitura dos Proxys quando forem guardados com sucesso;
  - Gerir ligações da base de dados central.

- Funcionalidades de comunicação com o exterior:
  - Existência e utilização de uma API de envio de emails e notificações Prowl para iPhone.

- Funcionalidades de logging:
  - Existência de um ficheiro Logger de Ações e Exceções.

Requisitos não funcionais

Os requisitos não funcionais a seguir apresentados descrevem aspetos relacionados com segurança e escalabilidade no contexto da plataforma MeWaGo, no âmbito deste estágio:

- Segurança:
  - Confidencialidade: existência de palavra-chave na autenticação que deverá ser encriptada com um algoritmo de encriptação de dados.

- Escalabilidade:
  - O servidor deverá suportar aquisição de dados de proxys na escala das dezenas sem existir perda de dados e baixa performance;
  - O servidor deverá suportar acesso de clientes na escala das dezenas, sem existir perda de dados e baixa performance.

Relativamente a segurança, o servidor será encarregue de processar informação de ínicio de sessão por parte de um cliente e verificar os seus dados de credenciais e respetivas permissões. O resultado final deverá ser transparente e correto para o utilizador final. No caso de um registo de utilizador, deverá ser guardado na base de dados um valor encriptado de password. Este é o resultado de uma *hash* aleatória encriptada com o algoritmo MD5

somada com a palavra-chave encriptada também com o algoritmo MD5. São criados pelo menos 10 casos de teste para verificar a encriptação na inserção de um novo utilizador na base de dados.

Quanto a escalabilidade, são criados 10 registos na base de dados para efeitos de teste que representam os proxys e outros 10 que representam clientes da aplicação desktop, para poderem aceder paralelamente e verificar o comportamento da performance.

4.1.1.2. Proxy

Requisitos funcionais

- Funcionalidades Core:
  - Mecanismo de sincronização;
  - Temporizador para correr todas as Drivers associadas ao Proxy e recolher valores de leitura de 15 em 15 minutos;
  - Carregar/atualizar ficheiros de configuração;
  - Tratamento de exceções.

- Funcionalidades de conecção:
  - Gerir ligações ao servidor;
  - Enviar valores das leituras;
  - Pedir valores para recolha de informações de configuração;
  - Avisar da atualização do servidor.

- Funcionalidades de manipulação de dados da base de addos:
  - Gerir ligações da base de dados local;
  - Guardar valores das leituras e apagar quando o servidor os guardar na base de dados central.

- Funcionalidades de logging:
  - Existência de um ficheiro Logger de Ações e Exceções.

Requisitos não funcionais

Os requisitos não funcionais a seguir apresentados descrevem aspetos relacionados com performance e escalabilidade, no contexto da plataforma MeWaGo, no âmbito deste estágio:

- Performance:
  - As tentativas de leitura de registos de dispositivos associados a um proxy e sua inserção na base de dados deverão demorar cerca de 1 minuto no máximo;
  - O tempo das leituras de todos os pontos de medição de um proxy deverá ser no máximo até 15 minutos.

- Escalabilidade:
  - O proxy deverá suportar acessos na escala das dezenas a dispositivos com centenas de canais no total, sem existir perda de dados e baixa performance.

De modo a testar estas funcionalidades, basta apenas colocar o *backend* em execução durante um dia, no mínimo, até 7 dias, para verificar o seu comportamento. São registados tempos

de leitura de registos e tempo de tentativas, se for o caso, de modo a verificar a validade dos requisitos de perfomance mecionados.

Quanto à escalabilidade, são criados no mínimo de 10 dispositivos com 10 canais cada, para efeitos de teste e observar o comportamento do proxy na aquisição de todos os valores correspondentes, num máximo de 15 minutos.

4.1.1.3. Drivers

Requisitos funcionais

- Funcionalidades Core:
    - Receção de número idêntico de parâmetros;
    - Recolher leituras do equipamento;
    - Tratamento de exceções.
- Funcionalidades de logging:
    - Existência de um ficheiro Logger de Ações e Exceções.

Requisitos não funcionais

Não são do âmbito deste estágio.

4.1.2. Frontend

Nesta secção pretende-se descrever a análise de requisitos realizada para o *frontend*. Foi feito um levantamento dos requisitos das necessidades do projeto anterior e juntei novas funcionalidades requeridas que faziam sentido para este projeto.

Uma vez que a plataforma irá permitir visualizar informação relativa a diversas grandezas, segundo usos específicos, tais como visualização de consumos, custos e impactos ambientais relacionados, será necessário repartir o mesmo em três ramos distintos. A estes tinha já sido dado o nome de "Vistas", das quais faziam parte as seguintes três: gestão técnica, gestão financeira e sensibilização ambiental. Eu propus a existência de uma secção com requisitos em comum e requisitos para uma quarta vista: gestão administrativa (*backoffice*). São de seguida apresentados os requisitos funcionais e não funcionais de alto nível para as duas componentes do *frontend*.

4.1.2.1. MeWaGo iOS

Nesta secção é descrito o processo de análise de requisitos realizado no contexto do desenvolvimento da aplicação iOS. A listagem de requisitos efetuada de seguida advém da necessidade das funcionalidades a desenvolver refletirem as exigências mais básicas para um lançamento de uma primeira versão de Alfa desta aplicação. Para já, apenas se deu importância a requisitos funcionais. Os seguintes requisitos representam as funcionalidades base que retiramos no contexto da análise de requisitos efetuada para a plataforma MeWaGo.

Requisitos funcionais

- Definição de preferências;
- Visualização de informação de um edifício;

- Visualização de localizações monitorizadas de um edifício;
- Seleção de grandezas a monitorizar;
- Seleção de tipo de períodos a monitorizar;
- Seleção de períodos a monitorizar;
- Seleção de localizações a monitorizar;
- Visualização do mapa de consumos;
- Visualização da informação da aplicação.

Requisitos não funcionais

Não são do âmbito deste estágio.

4.1.2.2. MeWaGo Desktop

Requisitos funcionais

- Requisitos Comuns:
    - Autenticação no sistema;
    - Visualização e edição de perfil de utilizador;
    - Visualização de valores instantâneos em tempo real;
    - Visualização de informações de edifícios e sua monitorização;
    - Geração de relatórios técnicos e financeiros.

- Gestão Administrativa:
    - Gestão de utilizadores e suas permissões, entidades, edifícios, localizações, dispositivos, canais e alarmes;

- Gestão Técnica:
    - Visualização de consumos gerais;
    - Visualização de consumos desagregados por localização, dispositivo e período de tempo;
    - Visualização de indicadores de consumos e sua estimativa ao longo do tempo;
    - Visualização de histórico de comparações de consumos;
    - Parametrização e notificação de alarmes.

- Gestão Financeira:
    - Visualização de custos gerais;

Requisitos não funcionais

Os requisitos não funcionais a seguir apresentados descrevem apenas aspetos relacionados com segurança, no contexto da plataforma MeWaGo, no âmbito deste estágio:

- Segurança:
    - Autenticação: autenticação, utilizando no mínimo uma combinação do nome de utilizador e palavra-chave;
    - Autorização: baseado no tipo de permissão, o utilizador poderá aceder a diferentes tipos de vistas/módulos;

De modo a testar estas funcionalidades, serão efetuados testes específicos relacionados com os dois itens mencionados acima. A base de dados é populada com valores de teste para utilizadores dos tipos: administrador, técnico e financeiro. Consoante o tipo de permissão dado pelo nome de utilizador e palavra-chave, deverá ser visualizado o separador ou informação correspondente ao utilizador, após tentativa de ínicio de sessão.

4.2. Arquitetura

Após a definição dos requisitos de um sistema, deverá partir-se para a construcção de uma arquitetura que espelhe as funcionalidades a que o mesmo se deverá encontrar preparado para responder. A definição correta de um arquitetura permite reduzir a complexidade desse sistema, tornando possível a divisão do mesmo em pequenos módulos. Esta redução de complexidade, conseguida através da abstração e separação da arquitetura em módulos mais pequenos, permite tornar a sistema MeWaGo num sistema modular e por conseguinte reutilizável.

Um risco grande que poderia levar ao insucesso do produto fnal, prendia-se com o facto de uma má definição da arquitetura. Daí ter-se procedido a ajustes durante o ínicio do segundo semestre para melhorar a arquitetura definida no primeiro semestre. Este foi um processo natural, que poderá acontecer a qualquer altura quando existe incoerência entre o conjunto de funcionalidades e o tipo de resposta que a arquitetura permite dar. Estas alterações na remodelação da arquitetura permitem atingir um modelo final melhorado. A sua alteração poderá ocorrer também devido a fatores externos, como por exemplo recorrer ao uso de diferentes tecnologias das que inicialmente foram escolhidas. Poderá assim ser necessário recorrer a uma abrupta mudança na decisão da estrutura da arquitetura de um sistema.

O resultado final com maior detalhe desta fase pode ser consultado nos Capítulo 2 e 3 do documento de Arquitetura de Software [Anexo B]. Este documento foi elaborado utilizando uma fusão entre os *templates* da IEEE e da RUP, disponibilizado para este tipo de documentos[49][50]. É de seguida apresentada a modelagem da arquitetura do sistema, apresentada segundo os Modelos Rational Rose e UML, utilizando o software Visual Paradigm Suite 5.1 (Visual Paradigm for UML 8.1 Enterprise Edition), com a licença Standard Edition Academic License[51].

4.2.1. Representação Primária

A Figura 4 – Arquitetura Geral da plataforma MeWaGo representa a arquitetura geral da plataforma MeWaGo, onde se pode distinguir facilmente as diferentes áreas modulares da mesma.

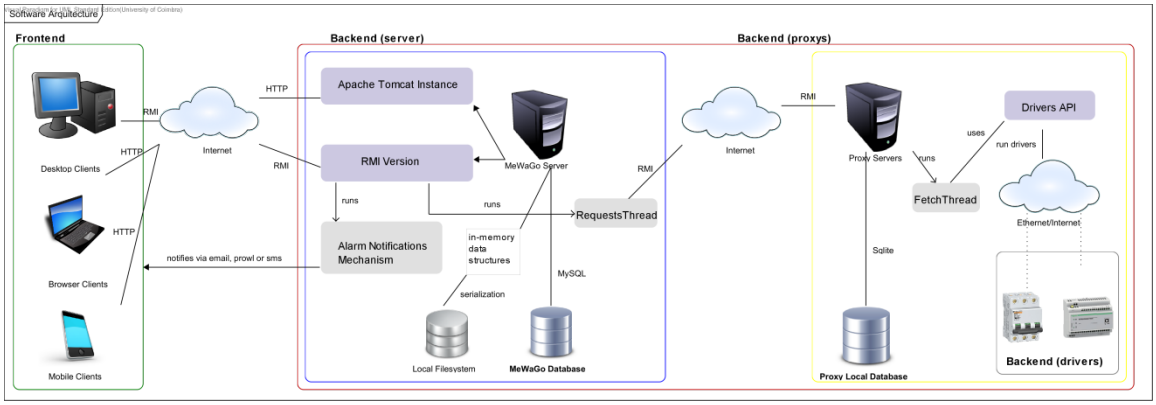

Figura 4 – Arquitetura Geral da plataforma MeWaGo

Esta definição de arquitetura permitiu subdividir o problema em vários mais pequenos e modulares. Num panorama geral, foram identificadas duas componentes: *backend* e *frontend*. A área a verde representa o *frontend* (clientes desktop, web e mobile), enquanto que a área vermelha representa o *backend*. Este último foi desagregado, dando origem a um servidor central (área a azul) e vários servidores locais ou proxys (área a amarelo). A componente do *frontend* não irá ser tratada neste capítulo, sendo mais tarde, no Capítulo 5, analisada com maior cuidado, relativamente ao produto final. Apenas é necessário reter a informação de que clientes desktop efetuam chamadas em RMI para conecção com o servidor, e clientes web e mobile utilizam o browser para visualizar informação adquirida por chamadas HTTP a um servidor Apache Tomcat existente na máquina que representa o servidor MeWaGo. Todo o processo de funcionamento do sistema MeWaGo e seus componentes, mediante a arquitetura apresentada, é explicado a fundo no documento de Arquitetura de Software [Anexo B].

4.2.2. Estrutura Interna

Irá ser agora analisada mais a fundo a arquitetura, onde se poderá distinguir os vários módulos necessários ao correto funcionamento do sistema, que permitem uma reutilização futura, um dos objetivos da implementação do software. A Figura 5 representa todos os elementos constituintes na arquitetura interna da plataforma MeWaGo. Uma explicação e os respetivos diagramas de classe das componentes do *backend* poderá ser consultada no Capítulo 2 do documento de Arquitetura de Software [Anexo B].

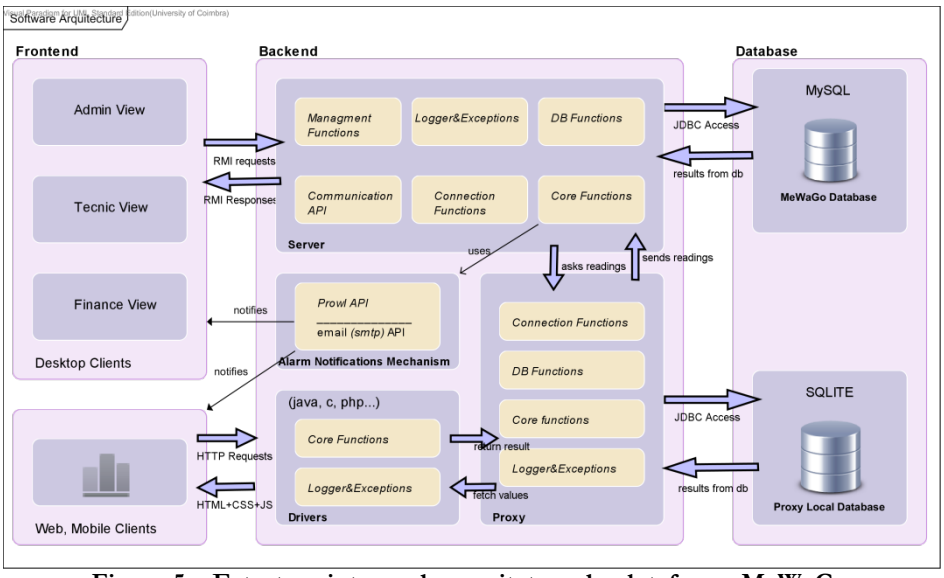

Figura 5 – Estrutura interna da arquitetura da plataforma MeWaGo

Foi tido o cuidado de utilizar SDP[53] em cada componente desenvolvida, como por exemplo o MVC[60] para o *frontend*. Este tipo de padrão de arquitetura permite o desenvolvimento em camadas modulares. O mesmo separa logicamente as diferentes partes do sistema e permite diminuir a sua complexidade, garantindo maior facilidade nas tarefas de manutenção e alterações do código.

4.2.3. Tecnologias e Ferramentas Utilizadas

Na Tabela 2 – Tecnologia Utilizada no MeWaGo são enumeradas as tecnologias e ferramentas base necessárias para o desenvolvimento da plataforma MeWaGo. Uma

descrição mais detalhada poderá ser visualizada no Capítulo 3 do documento de Arquitetura de Software [Anexo B]:

Tecnologia Utilizada	Componentes	Descrição/Justificação
Linguagem Java	Backend (Server, Proxy e Drivers) e Frontend (MeWaGo Desktop)	- Linguagem para programação das aplicações Server, Proxy, Drivers e MeWaGo Desktop. - Permite portabilidade de aplicações para qualquer plataforma. - Maior flexibilidade para integração com APIs, por exemplo de notificações de alarmes via email, sms ou prowl e de geração de gráficos. - Utilizado para desenvolvimento de drivers de acesso a equipamento de monitorização energética. - Utiliza *garbage collector* para limpeza automática de recursos que não estão a ser utilizados. - Tecnologia que permite interligar toda a tecnologia necessária a utilizar neste projeto. - Fácil utilização pelo estagiário, maximizando o desempenho, uma vez que não será necessário aprender novas linguagens que impliquem demasiado tempo. - É necessário testar também este tipo de tecnologia nunca antes utilizada neste projeto.
Java RMI	Backend (Server e Proxy) e Frontend (MeWaGo Desktop)	- Tecnologia utilizada para comunicação através da Internet, sob os protocolos TCP/IP, para comunicação entre o Server-Proxy e vice-versa e entre o Server-MeWaGo Desktop e vice-versa. - Interface de programação que permite a chamada de métodos remotos, utilizada de forma bastante fácil, simples e transparente.
Eclipse Helios for Java Developers	Backend e Frontend	- Ambiente para desenvolvimento de aplicações que utilizam tecnologia Java.
Linguagem C	Drivers	- Utilizada para criação de drivers de acesso a equipamento de monitorização energética.
Sqlite	Database	- Biblioteca que não necessita de configuração e livre para utilização comercial; - Utilizada na base de dado de proxys.
MySQL Workbench 5.2. CE	Database	- Software para modelação de base de dados, desenvolvimento SQL e com ferramentas administrativas para configuração de servidores, utilizadores e utilizadores.
HTML, CSS, JS, PHP	Frontend	- Tecnologia web utilizada para criação de páginas web (gráficos para frontend). - Javascript é utilizado para geração de gráficos. - PHP é utilizado na criação de scripts e funções de acesso à base de dados para criação de ficheiros CSV que dão suporte à geração de gráficos. - PHP é também utilizado para desenvolvimento de Drivers de acesso a equipamento de monitorização energética. - HTML é utilizado para apresentação de páginas web, juntamente com CSS que permite definir estilos personalizados de páginas.
Bitnami WAMP Stack	Frontend	- Integra Apache HTTP Server, MySQL e PHP;

		- Permite hospedagem de páginas web para testes locais do dashboard do DEEC.
amCharts bundle	Frontend	- Permite a geração de gráficos javascript para integração no *frontend.*
xCode IDE	MeWaGo iOS	- Ambiente de desenvolvimento em Mac OS para dispositivos da Apple iOS.

Tabela 2 – Tecnologia Utilizada no MeWaGo

4.2.4. Modelo de Dados

Uma vez que existirão dados a serem salvaguardados para uso futuro no sistema, os mesmos terão que fazer parte de um mecanismo de persistência que permita guardar estes mesmos dados eternamente para qualquer tipo de uso. Para tal foi elaborado um modelo de dados com as entidades necessárias, isto é, com as tabelas e seus atributos e relações entre elas que representam todos os dados a serem analisados e manuseados no sistema e pelo mesmo.

Esta secção encontra-se especificada no Capítulo 3.4 do documento de Arquitetura de Software [Anexo B]. O mesmo contém o modelo de dados, representado por um diagrama E-R, elaborado no MySQL Workbench 5.2 CE que carateriza o modelo físico de dados adotado para o sistema desenvolvido.

O modelo de dados utilizado para este protótipo de aplicação iOS para o projeto MeWaGo teve como alicerce a base de dados rudimentar do servidor do projeto antigo E-Monitor. Foi utilizada a linguagem PHP para acesso a dados da base de dados existente no servidor do E-Monitor.

Capítulo 5.
Trabalho Realizado

Neste capítulo apresento todo o trabalho desenvolvido no sentido de cumprir com os objetivos deste projeto. É abordado o progresso de estágio desde o seu ínicio no primeiro semestre até ao progresso durante o segundo semestre. O primeiro semestre compreende desde o estudo do conceito da plataforma MeWaGo até a construcção de um simulador para carregamento de dados. Todo o produto de implementação, resultado do segundo semestre, é visto nas sub-secções seguintes.

5.1. Progresso no primeiro semestre

5.1.1. Estudo do Conceito da Plataforma

Inicialmente, na primeira etapa distinguida como Fase 1 do plano de estágio, foi necessário proceder a um estudo prévio do conceito da plataforma. Nesta fase contribuí na escrita de um documento de apresentação do projeto MeWaGo para o concurso lançado pela PT Galp, denominado INNOVATION Challenge. A primeira fase do concurso consistia na elaboração de um documento com o conceito e principais funcionalidades da aplicação MeWaGo para ambiente móvel. A elaboração deste documento permitiu compreender melhor o conceito da plataforma a nível de funcionamento e decidir qual o dispositivo móvel a utilizar para implementação. A segunda fase compreendia o desenvolvimento de uma demonstração para o MeWaGo iOS, integrada na Fase 5 do plano de estágio.

5.1.2. Análise de Padrões de Consumos

Esta componente foi integrada na Fase 3 do plano de estágio, tendo como pano de fundo o estudo prévio do conceito da plataforma e a investigação sobre material fornecido pelo orientador da empresa para esse efeito. Esta tarefa foi realizada para poder compreender todas as entidades envolvidas e necessárias para criação do novo modelo de dados da plataforma a desenvolver. Estes padrões de consumo analisados são referentes a um caso de estudo específico a descrever de seguida.

5.1.2.1. Caso de Estudo (DEEC)

Uma vez que o DEEC foi o berço de incubação do projeto E-Monitor, dando mais tarde lugar ao projeto MeWaGo, foi este o caso de estudo alvo. É por isso importante descrever um pouco alguma informação relacionada com o mesmo, em termos de consumo energético.

Como suporte ao conceito deste projeto, foram recolhidas nesta atividade informações e dados úteis de modo a efetuar uma caracterização detalhada deste edifício. O objetivo passou por demonstrar a utilidade da aplicação de padrões de consumo, fruto dos seus gastos energéticos excessivos. Desta maneira, foi possível fundamentar de um modo, não só teórico, como também experimental, o projeto em causa. Este teve como foco principal, essencialmente uma análise mais geral dos consumos deste edifício no ano letivo de 2009/2010.

O edifício em estudo foi caracterizado tendo em conta os regulamentos energéticos e o perfil de utilização. Com os dados de energia elétrica provenientes desse estudo e respetivos DCs, foi possível examinar o impacto significativo da presença de utilizadores no espaço e das influências meteorológicas nas alterações de consumos. A pesquisa destes dados serviu para um melhor conhecimento do comportamento energético do edifício. Foi por isso posta de parte a descrição detalhada dos aspetos de caraterização do edifício de estudo.

É de seguida apresentada a Figura 6 – DAC do DEEC em 2010 centrado no consumo médio, onde estão representados os consumos energéticos do ano de 2010 do DEEC que obedecem a determinados padrões, variando estes segundo diversos fatores.

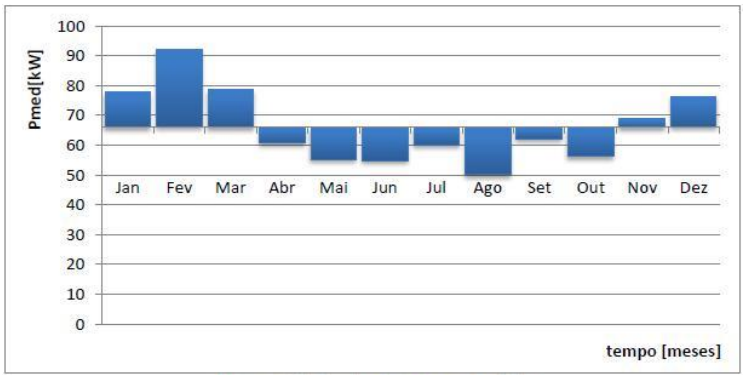

Figura 6 – DAC do DEEC em 2010 centrado no consumo médio

Com a figura anterior, por exemplo, pode-se verificar quais são os meses de maior consumo, onde as potências médias mensais são centradas na potência média anual, dando relevo aos meses que estão acima dessa média. São eles janeiro, fevereiro, março, novembro e dezembro. Poderá concluir-se também, após a comparação do comportamento entre os 12 meses existentes no DC ao longo do ano respetivo, quais os meses mais quentes e mais frios. Isto é possível, após verificar o valor das potências mínimas dos meses mais frios. Estes valores demonstram uma provável utilização de aquecimentos elétricos que poderão eventualmente ter sido deixado ligados durante a noite nos meses mais frios.

Uma análise mais detalhada destes consumos pode levar a um grande potencial de poupança energética, o que torna a intervenção da plataforma MeWaGo indispensável neste campo, pela sua importância e eficácia.

Este estudo teórico fundamentado permitiu a criação de um simulador, o mais próximo do nível exigido pelo plano de estágio, que permitiu a criação de padrões de consumos. Os mesmos respeitam alguns fatores importantes a serem descritos de seguidas.

5.1.2.2. Simulador de Tráfego

Numa terceira fase do primeiro semestre desenvolvi um pequeno simulador de tráfego para carregamento de dados na base de dados, o que irá permitir futuramente a inclusão de novos dispositivos de forma mais fácil. Esta atividade foi baseada no modelo de dados criado e foi levada a cabo após ter sido feito o *dumping* desta.

Conforme exigido pela empresa, foi necessário implementar um simulador que teria que satisfazer as seguintes necessidades:

- Carregamento de dados reais e para efeitos de teste na base de dados;
- Inclusão flexível de novos dispositivos.

O carregamento de dados reais e de teste na base de dados correspondem à criação de *scripts* em formato sql que garantem a inserção de valores na base de dados relacionados com o calendário anual no período de 10 ou mais anos, perfis semanais e anuais de consumo e tarifas de teste. Constou também informações de edifícios, alguns utilizadores e suas permissões, entidades, edifícios, localizações, dispositivos, canais e grandezas a medir para efeitos de teste.

Desenvolvi este simulador com a utilização de SDP[53], o que permitiu uma maior flexibilidade na programação do mesmo e futura utilização para *debugging* mais fácil ou, porventura, para a adição de novos módulos ou alteração dos já existentes. Foi preciso também gerar a documentação javadoc[54] [55] disponível. [55]

Este simulador foi criado para permitir a sua integração com o *backend* desenvolvido. O mesmo utiliza comandos adicionados como argumentos para executar as diferentes vertentes de simulação – estática ou dinâmica – ou poderão ser efetuadas chamadas a classes geradoras ou simuladoras por parte do servidor.

Poderão, através de comandos, ser lidos ficheiros de texto, carregados valores de teste ou da base de dados. Poderão ser gravados valores em ficheiros sql que servirão como *scripts* de *dumping* da mesma. Por outro lado existem também classes específicas que permitem integrar de uma maneira bastante eficaz vários controladores específicos responsáveis pelas leituras de diferentes dispositivos.

Desta maneira foi desenhado um simulador que pode receber dois comandos: *static* e *dynamic*. Estes comandos podem receber vários parâmetros que facilitam a execução dos módulos acima explicados. Para um esclarecimento sobre o seu uso para a empresa foi elaborado um manual de comandos deste simulador.

Este simulador permitiu a geração de valores de leituras energéticas durante o período do ano de 2011 e de 2012 (até fevereiro), e sua inserção na base de dados central do servidor MeWaGo. A implementação deste simulador deu abertura a uma evolução do mesmo que gerou o produto final referente ao Proxy.

5.2. Progresso no segundo semestre

5.2.1. Servidor (RMI)

Esta componente importante do *backend* foi desenvolvida na sua plenitude no tempo disposto, tendo sido produzido um produto final na versão Release Candidate esperada. Foi utilizada uma máquina do DEEC com o sistema operativo Linux CentOS 6.2[56], para utilização futura do projeto MeWaGo, tendo sido usados para acesso a esta os clientes de SFTP, FTP e SSH WinSCP[57] e o Putty[58] para ambientes Windows. Após alguns testes iniciais, quando o *backend* se encontrava preparado para *deploy* nesta máquina, foi realizado o levantamento de requisitos a nível de software necessário para correr, tanto o servidor com o proxy do *backend* do projeto. Todo este processo originou um documento detalhado com toda a preparação de instalação, *deploy* e configuração, entre outros aspetos bastante importantes no manuseamento desta aplicação para ambientes Linux. Dos componentes importantes mais necessários para a execução deste servidor, foi instalado o Java e MySQL para *dumping* da base de dados central. Este serviço começa por sincronizar com o relógio atual e, para além de tratar de pedidos de aplicações cliente MeWaGo, possui uma *thread* que entra em rotina de 15 em 15 minutos para comunicar por RMI com todos os proxys que se

encontram ativos no momento e vai buscar todos os valores lidos. Após armazenados com sucesso na base de dados central, o servidor ordena a remoção dos valores correspondentes dos proxys e entra de novo na rotina normal. A Figura 7 demonstra o processo de execução do serviço do servidor a correr como um processo na máquina Linux, estando constantemente a gerar *loggers*. Apesar de ter sido testado o cenário de um servidor a descarregar dados de dois edifícios (DEEC e DEM), apenas é mostrada na seguinte figura a leitura de treze pontos de leitura do DEEC.

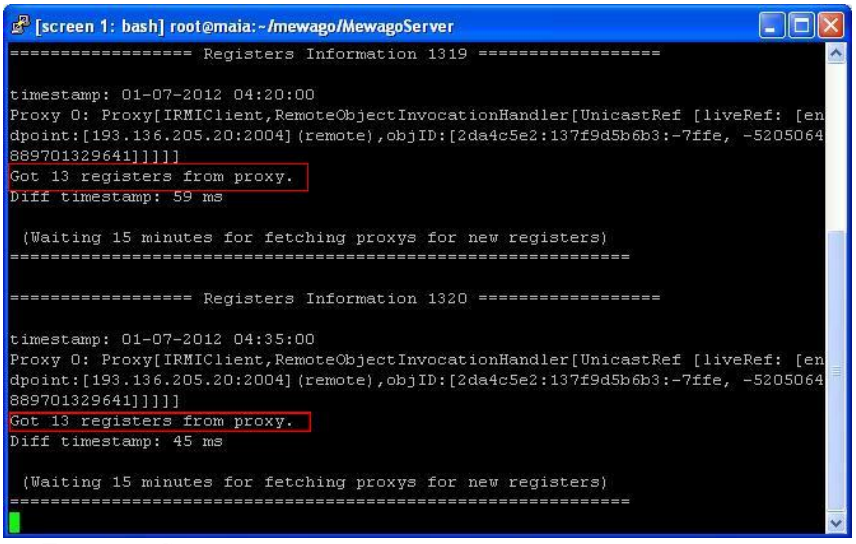

Figura 7 – Leitura de proxys e armazenamento de dados do servidor

O servidor está preparado para cenários de falhas, isto é, se estiver embaixo, independentemente do tempo decorrido, quando volta ao ativo o procedimento é normalizado. Isto é, um ficheiro de *backup* contém todas as ligações pendentes com os proxys e no ato de iniciação o servidor lê este ficheiro. Quando entra na rotina de ir buscar valores aos proxys, os valores locais que podem ter sido obtidos em intervalos de tempo maiores de 15 minutos são guardados. Não existe por isso perda de dados em caso de falhas do servidor.

5.2.2. Servidor (Apache Tomcat)

Foi necessário instalar o Apache Tomcat na máquina na qual se estava a trabalhar para poder hospedar páginas HTML e PHP para visualização de gráficos da amCharts. Assim foi possível dar resposta a vertentes web de aplicações cliente para demonstração dos gráficos criados. Este servidor, em fase final de implementação, serviu também para hospedar uma pequena plataforma (dashboard) de consumos do DEEC que irá ser foco de detalhe mais à frente.

5.2.3. Proxy

Esta segunda componente do *backend* foi implementada com todas as funcionalidades necessárias no tempo limite, tendo sido produzido um produto final na versão Release Candidate esperada. Esta componente é o resultado da evolução da implementação do simulador de tráfego. O proxy representa um edifício, sendo por isso necessário instalar esta versão em vários edifícios para poder usufruir da funcionalidade da plataforma MeWaGo de monitorizar consumos de vários edifícios. Depois de instalado o proxy é necessário preencher a base de dados central do edifício com os dados associados a esse edifício. Foram testados dois proxys diferentes que representaram o DEEC (edifício de caso de

estudo atual) e o DEM. Para o DEEC, foram instalados durante este ano mais alguns equipamentos de leitura de grandezas energéticas (eletricidade e temperatura), para além dos que já se encontravam em funcionamento de anos anteriores. A recolha de informação efetuada de pontos monitorizados neste semestre deu origem a um Mapa de Instalação Elétrica do DEEC que pode ser visualizado no documento de Mapa de Instalação Elétrica do DEEC [Anexo C]. A Figura 8 representa a leitura, de 15 em 15 minutos, dos vários pontos do DEEC, já com a representação de um canal virtual para a Torre T (esta expressão é o cálculo da diferença de consumos entre o consumo Geral do Edifício menos as restantes Torres e Anfiteatros).

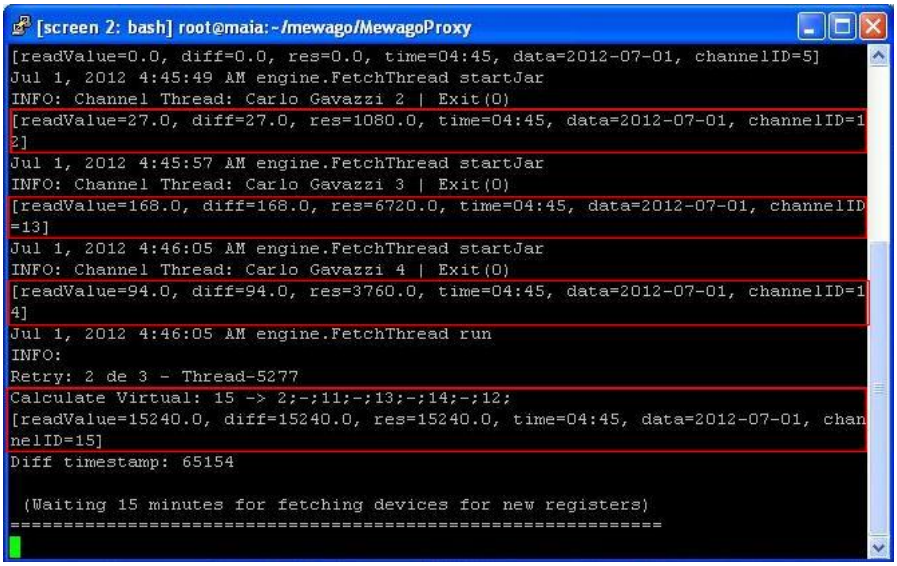

Figura 8 – Leitura de dados das drivers pelo proxy DEEC

5.2.4. Drivers

No decorrer do segundo semestre foi necessário implementar algumas drivers para três dispositivos diferentes. São eles o Zelio da Schneider, o iMeterBox da ISA e o USBAdapter da Usense. Respetivamente, foram utilizadas as linguagens c e java para os dois últimos. Para o iMeterBox era necessário apenas ler um ficheiro XML, enquanto que para o terceiro era necessário ler um valor em ficheiro de texto. O primeiro já se encontrava parcialmente implementado, pelo que foi apenas necessário adaptar consoante as novas necessidades no projeto atual. Estes drivers poderão ser utilizadas futuramente como modelo para desenvolver novas drivers de acesso a diferentes dispositivos com que se trabalhe futuramente, tendo sido por isso uniformizado o seu método de implementação – recebem o mesmo número de argumentos e imprimem um resultado apenas (valor correcto ou erro).

5.2.5. Trabalho Extra

Este capítulo especifica todo o trabalho de implementação que resultou em produtos finais de versões Alfa e Beta.

5.2.5.1. Gráficos

Foi utilizado a biblioteca amCharts para criação de inúmeros gráficos, mais focados nos consumos do que nos custos para utilização futura. Este trabalho não constava no plano inicial de estágio mas serviu para utilização e adaptação ao *frontend* atualmente implementado para a aplicação MeWaGo Desktop. Todos os gráficos implementados poderão ser

visualizados no documento de Gráficos do MeWaGo [Anexo D]. A Figura 9 representa alguns dos gráficos implementadas que foram utilizados no *frontend*, nomeadamente no dashboard do DEEC, MeWaGo Desktop e MeWaGo iOS.

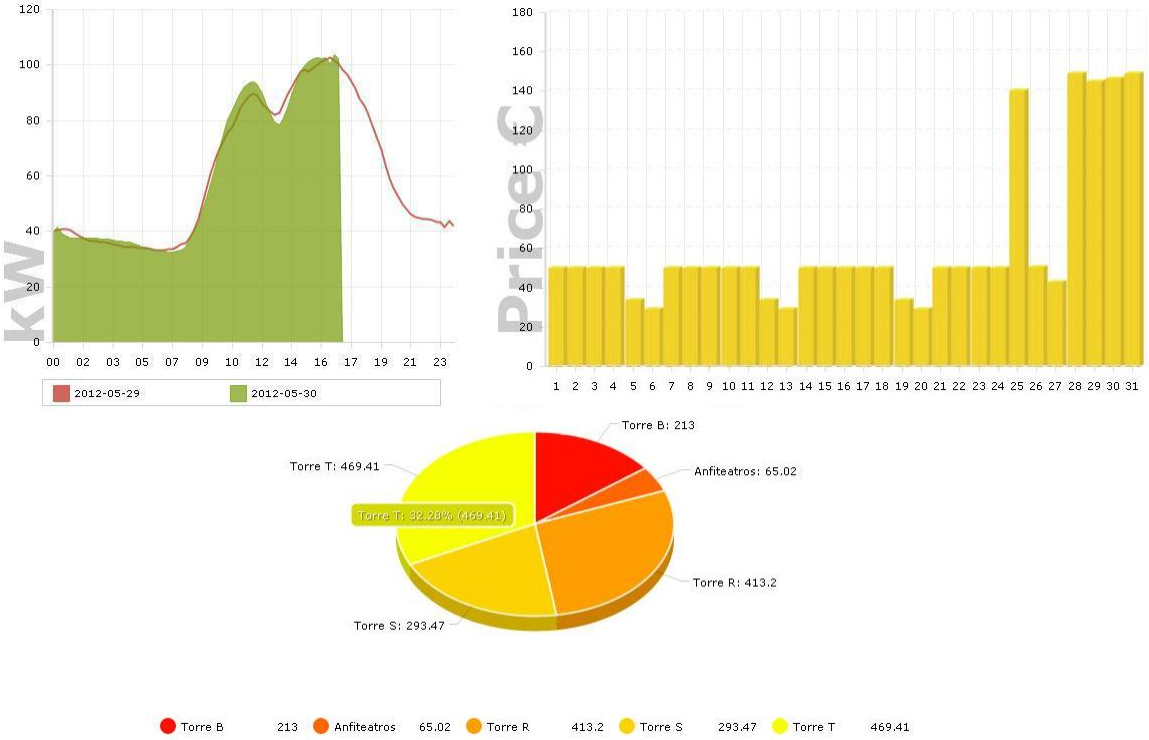

Figura 9 – Gráfico de consumos e custos de eletricidade geral do DEEC

A máquina que representa o servidor para o projeto MeWaGo detém os ficheiros PHP que utilizam javascript para geração de gráficos. Inicialmente são efetuadas chamadas a métodos de um ficheiro PHP geral que permite gerar dados obtidos da base de dados central MeWaGo. Existe um ficheiro para geração de dados de consumo e outro para geração de dados de custos. Após criados os ficheiros, o javascript trata de os ler e criar o gráfico respetivo. O URL serve para reencaminhar para os ficheiros certos e o PHP trata de ler os parâmetros necessários para a geração do gráfico em questão. Esses parâmetros permitem manipular a visualização de gráficos, mediante as necessidades apresentadas.

5.2.5.2. Dashboard DEEC

Os restantes elementos da equipa MeWaGo chegaram à conclusão que os gráficos implementados poderiam dar início a uma versão Beta de um protótipo para a plataforma web. Para tal, foi utilizado o Symfony2[59] para criação de um dashboard próprio para cada departamento. Este foi desenvolvido utilizando os gráficos implementados apenas para dar resposta as necessidades momentâneas do DEEC. A Figura 10 – Cabeçalho do dashboard do DEEC representa o cabeçalho das páginas HTML. Existe um menu para ver consumos desagregados por localização e consumos da localização escolhida desde a meia-noite até ao último valor lido do dia corrente. Existem *gauges* com os últimos valores lidos para as localizações desagregadas da localização geral, para a eletricidade. Gráficos com comparações ao dia homólogo e à semana anterior com a atual fazem-se acompanhar com tabelas que detém valores importantes para visualização. Alguns gráficos circulares importantes são também utilizados para desagregação de valores de consumos semanais, bem como os indicadores anuais para as várias semanas e dias de um ano e sua projeção. Durante a fase de implementação desta vertente, todos os elementos estiveram presentes e

contribuíram ativamente. Foi utilizado o padrão de desenvolvimento de arquitetura MVC[60], o que permitiu criar código estruturado e modular. Torna-se por isso mais fácil a adição de funcionalidades ou alteração das já existentes.

Figura 10 – Cabeçalho do dashboard do DEEC

A Figura 11 representa uma secção do dashboard do DEEC com utilização de alguns gráficos criados através da amCharts.

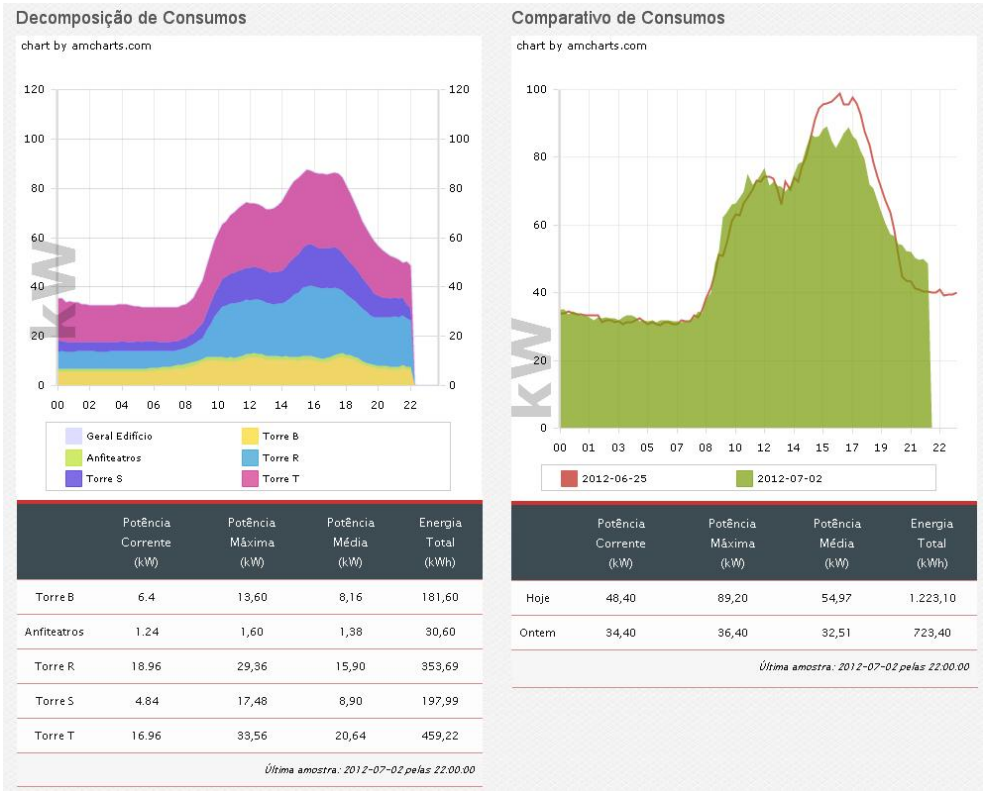

Figura 11 – Gráficos diários de consumos desagregados de eletricidade geral do DEEC

5.2.5.3. Mewago Desktop

Para o *frontend* da plataforma MeWaGo, optou-se por desenvolver uma versão Beta da aplicação para versão desktop em java. Praticamente todas as funcionalidades administrativas foram implementadas e para visualização de gráficos, utilizou-se um browser integrado em java para acesso aos links que mostram os gráficos previamente construídos com a amCharts. Na Figura 12 – MeWaGo Desktop pode-se visualizar o separador do módulo técnico do produto final. Aqui são visualizados os últimos valores diários, semanais, mensais ou anuais, conforme a escolha de filtragem do utilizador. Podem ser visualizados gráficos de diferenças, desagregação por áreas, localizações ou períodos horários, respetivos indicadores e download do ficheiro CSV que possui os dados numéricos necessários para construir o gráfico visualizado, bem como capturar uma imagem JPEG do gráfico para consulta futura, ou geração de relatórios.

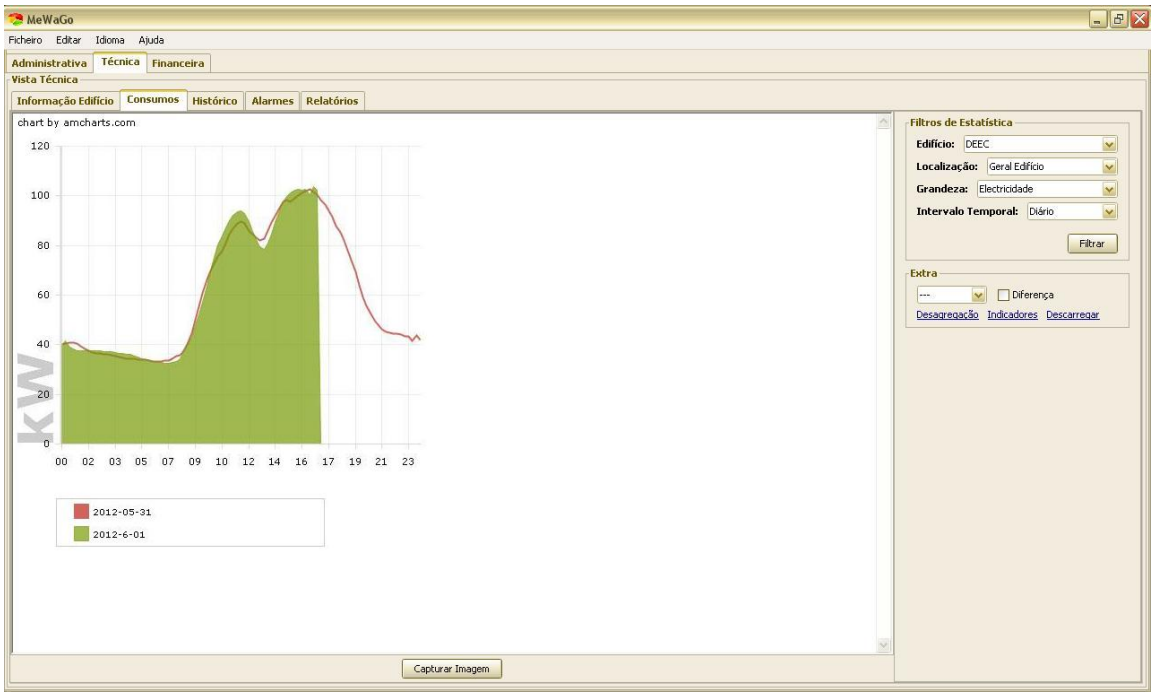

Figura 12 – MeWaGo Desktop

5.2.5.4. Mewago iOS

A aplicação MeWaGo iOS foi implementada na íntegra, consoante os requisitos funcionais propostos pela empresa, tendo sido conseguido finalizar o produto numa versão Alfa. Uma vez que só é possível desenvolver aplicações para a plataforma iOS com o iOS SDK e o IDE xCode em ambiente MAC OS X, foi-me fornecido um computador com o sistema operativo adequado e as ferramentas necessárias. A plataforma xCode possui já um simulador nativo de ambientes iOS, o que permitiu um desenvolvimento mais próximo do real da aplicação. Existe ainda um portal para programadores, Dev Center[52], que contém uma infinidade de recursos, incluindo vídeos, código fonte e documentação técnica. A Figura 13 – MeWaGo iOS mostra três das interfaces nativas do iPhone que são utilizadas na aplicação, para visualização de informação de um edifício ao qual o utilizador se ligou para monitorizar os seus consumos, escolha do período de consumo (eletricidade mensal) e o respetivo mapa de consumos, utilizando uma página com um gráfico criado com a biblioteca amCharts numa fase inicial de testes.

Figura 13 – MeWaGo iOS

5.2.5.5. Monitor DEEC - Vertente de Sensibilização

Foram também aproveitados os gráficos e tabelas criados durante o segundo semestre para permitir a visualização de consumos num monitor que irá localizar-se no hall de entrada do DEEC. Esta componente irá permitir abertura a uma evolução numa próxima vertente de sensibilização para o projeto MeWaGo. A Figura 13 – MeWaGo iOS corresponde a uma fotografia tirada ao monitor que contém o conteúdo, ainda em fase de testes, a ser visualizado no DEEC.

Figura 14 – Monitor a utilizar no DEEC com gráficos de consumos

5.2.5.6. Documentos e Manuais

Foram elaborados alguns documentos paralelamente à tarefa de implementação de cada componente e finalizado no final de cada módulo. São estes:

Manual de Utilizador Mewago iOS

Contém a descrição de todas as funcionalidades com o objetivo de guiar um potencial utilizador desta aplicação para a correta operação e execução de comandos, entre outros, de

modo a satisfazer as suas necessidades e exigências. Este documento pode ser consultado no Manual de Utilizador iOS [Anexo F].

Manual de Utilizador Mewago Desktop
Descreve todas as funcionalidades com o objetivo de guiar um potencial utilizador para a correta operação e execução de comandos de modo a satisfazer as suas necessidades. Este documento pode ser consultado no Manual de Utilizador MeWaGo Desktop [Anexo G].

Manual de Programador
Este manual é direcionado para um programador ou programadores futuros que necessitem de informação do modo de funcionamento de cada módulo do software para adição ou alteração de funcionalidades ou mesmo para a compreensão de todo o processo.

Manual de Instalação e Configuração
Aqui é descrito todo o processo de instalação e configuração dos vários componentes do software da plataforma MeWaGo. Este documento foi fruto de um trabalho extenso e elaborado desenvolvido ao longo de todo o segundo semestre. Possui também uma lista de problemas e soluções encontrados. Num panorama geral, estes podem ser vistos no Capítulo 6 do documento de Planeamento [Anexo E]. Assim como riscos inerentes ao projecto que poderão ser encontrados no Capítulo 5 do mesmo documento.

Documentos de Plano de Testes e Casos de Teste
Nestes documentos, elaborados tanto para o *backend* como para o *frontend*, constam todos os casos de teste construídos para validar o funcionamento de todo o trabalho desenvolvido. Os testes funcionais e de usabilidade das funcionalidades da aplicação iOS encontram-se no documento de Testes MeWaGo iOS [Anexo H]. Alguns testes unitários e de sistema às funcionalidades gerais da plataforma encontram-se no documento de Testes MeWaGo Desktop [Anexo I].

Capítulo 6.
Conclusões e Trabalho Futuro

Neste capítulo final retirei algumas conclusões de todo o trabalho realizado durante o decorrer deste estágio. É também feita uma referência ao trabalho futuro a realizar no decorrer do segundo semestre.

6.1. Conclusão

O trabalho realizado ao longo deste ano letivo decorreu no âmbito do projeto MeWaGo cujo objetivo principal foi o desenvolvimento de uma plataforma de monitorização de consumos em edifícios. Era pretendido também a criação de protótipos de demonstração de algumas das funcionalidades a que a plataforma se encontrava preparada. Uma delas consistia numa aplicação para desktop, desenvolvida em java, com a maior parte das funcionalidades. Com a massificação dos *Smartphones* surgiu um novo mercado de oportunidades para as aplicações móveis. Desta forma, pensou-se num desenvolvimento adicional de uma aplicação cliente *user-friendly* que satisfizesse as necessidades mais básicas do cliente para a criação de um segundo protótipo para demonstração. Este exercício permitiu ao estagiário familiarizar-se com programação para iOS, aumentando a sua bagagem de conhecimento em tecnologias associadas, das quais nunca teve contacto. Foram também aprofundados conhecimentos sobre tecnologia web (PHP, HTML, CSS, Javascript) na implementação do dashboard para o DEEC.

Esta plataforma insere-se na área da eficiência energética e tem como objetivos principais a criação de uma ferramenta eficiente que permita aplicar políticas e adotar medidas de redução de custos, consumos e consequente emissão de CO2 para a atmosfera. Estes objetivos poderão ser realizados uma vez que esta ferramenta permite o controlo desses fatores e indicadores relacionados e facultação de parametrização e notificação de alarmes aquando da existência de elevados custos ou consumos.

Procedeu-se inicialmente a um estudo do estado da arte onde foi verificado a existência de alguns sistemas de monitorização concorrentes existentes nesta área. Este estudo serviu para compreender melhor as funcionalidades de um sistema desta envergadura, o que permitiu fazer uma comparação qualitativa entre essas aplicações e o MeWaGo, dando maior valor a funcionalidades interessantes que esta plataforma possui, que outras não o possuam.

Após um planeamento inicial cuidado das várias fases inerentes a este projeto de estágio e a seleção de uma metodologia ágil de desenvolvimento de software iterativa e incremental - SCRUM - procedeu-se a uma junção de um conjunto de tarefas que serviram de suporte à implementação do projeto durante o segundo semestre.

A metodologia adotada deu resposta às necessidades e exigências do tipo e nível de dificuldade e dimensão do projeto. Tanto a empresa, como o estagiário, não estavam familiarizados com a metodologia SCRUM, pelo que, ao aplicá-la com algum sucesso, foi possível aprender um pouco sobre a mesma, criando uma preparação repleta de informação e experiência com esta metodologia ágil.

As tarefas de levantamento de requisitos e elaboração do documento correspondente, bem como do documento respetivo da arquitetura com o design dos mockups para auxiliar na fase de implementação foram bastante úteis. Desta maneira foi possível realizar um plano de testes detalhado para mais tarde poder validar todo o trabalho desenvolvido durante o segundo semestre.

Foi construído um simulador de tráfego durante a primeira fase, tendo como suporte uma analise de padrões de perfil e o modelo de dados criado para validação de dados. Este trabalho contribuiu eficazmente para a continuação do desenvolvimento da plataforma MeWaGo durante o segundo semestre.

A título conclusivo, as tarefas realizadas auxiliaram na implementação da plataforma de monitorização de consumos em edifícios. Assim, foi possível desenvolver uma ferramenta que caminha no sentido de reduzir o impacto ambiental e custos acoplados a consumos energéticos em grandes edificações.

6.2. Trabalho Futuro

Após dado por concluído todo o trabalho realizado durante o segundo semestre com êxito, verificou-se que se deverá de futuro corrigir pequenos erros da aplicação MeWaGo Desktop. Apesar de esta aplicação não estar prevista inicialmente no plano de estágio, foi feito um esforço no sentido de a completar o máximo possível com as funcionalidades desejadas.

Em relação ao conjunto de gráficos necessários para representar consumos e custos, deverá de futuro, dar-se maior ênfase aos gráficos de custos, uma vez que não se encontram todos implementados. Estes irão contribuir para um plano futuro de desenvolvimento da plataforma web para o MeWaGo, ou porventura para adicionar informação importante no dashboard do DEEC ou de outros edifícios.

Todo este material organizado, à exceção do Servidor e Proxy, deverá ser alvo de discussão, correção e otimização, bem como a funcionalidade de alarmes da plataforma deverá ser alvo de atenção para expansão e melhoramentos.

Poderá dar-se continuidade à documentação existente e elaborar a que for sendo necessária no decorrer deste projeto até uma fase mais madura.

O *backend* desenvolvido, utilizando SDP, irá permitir uma fácil integração de novos módulos ou porventura, de implementação de novas *drivers* conforme forem surgindo novos equipamentos de leituras de grandezas energéticas. O mesmo está preparado para ser instalado em vários edifícios alvos de monitorização, dando asas para novas funcionalidades que usufruam destas capacidades do motor da plataforma desenvolvida.

Referências

[1]. IPN – Instituto Pedro Nunes. *Ideia Luminosa da FCTUC vence concurso nacional promovido pela EDP*. Disponível em: https://www.ipn.pt/si/event/dataNews.do?elementId=337. Data de acesso: 24-09-2011

[2]. ERSE – Entidade Reguladora dos Serviços Energéticos. *Portal ERSE – Bem-vindo ao portal ERSE*. Disponível em: www.erse.pt. Data de acesso: 24-09-2011

[3]. PER – Portal das Energias Renováveis. *Eficiência Energética em Edifícios. Disponível* em: http://www.eficiencia-energetica.com/html/eee/eee.htm. Data de acesso: 24-09-2011

[4]. EUR-Lex – Acesso ao direito da União Europeia. Comunicação da Comissão. *Plano de Ação para a Eficiência Energética: Concretizar o Potencial*. Disponível em: http://eur-lex.europa.eu/smartapi/cgi/sga_doc?smartapi!celexplus!prod!DocNumber&lg=pt&type_doc=COMfinal&an_doc=2006&nu_doc=545. Data de acesso: 24-09-2011

[5]. Europa – Sínteses da legislação da EU. *Plano de Ação para a Eficência Energética (2007-2012)*. Disponível em: http://europa.eu/legislation_summaries/energy/energy_efficiency/l27064_pt.htm. Data de acesso: 24-09-2011

[6]. PER – Portal das Energias Renováveis. *Sistema Nacional de Certificação Energética e da Qualidade do Ar Interior (SCE) Decreto-Lei n.º78/2006*. Disponível em: http://www.eficiencia-energetica.com/html/eee/eee_sce.htm. Data de acesso: 24-09-2011

[7]. PER – Portal das Energias Renováveis. *Regulamento dos Sistemas Energéticos de Climatização em Edifícios (RSECE) Decreto-Lei n.º 79/2006*. Disponível em: http://www.eficiencia-energetica.com/html/eee/eee_rsece.htm. Data de acesso: 24-09-2011

[8]. PER – Portal das Energias Renováveis. *Regulamento das Características de Comportamento Térmico dos Edifícios (RCCTE) Decreto-Lei n.º 80/2006*. Disponível em: http://www.eficiencia-energetica.com/html/eee/eee_rccte.htm. Data de acesso: 24-09-2011

[9]. The Modbus Organization. Disponível em: www.modbus.org. Data de acesso: 05-09-2011

[10]. Serial Protocols Compared. Disponível em: http://www.eetimes.com/design/embedded/4023975/Serial-Protocols-Compared. Data de acesso: 05-09-2011

[11]. Construção Sustentável. *Sistemas de Gestão Contínua e Monitorização Contínua* . Disponível em: http://construcaosustentavel.pt/index.php?/O-Livro-%7C%7C-Construcao-Sustentavel/Eficiencia-Energetica/Sistemas-de-Gestao-Continua-e-Monitorizacao-Continua. Data de acesso: 05-09-2011

[12]. Serious Energy Monitoring | EPA Benchmarking | Spectral Analysis | Budgeting and Chargebacks. Disponível em: http://www.seriousenergy.com/energy-management/applications/monitoring.html. Data de acesso: 26-06-2012

[13]. Home | Open Energy Monitor. Disponível em: http://openenergymonitor.org/emon/. Data de acesso: 26-06-2012

[14]. Opower. Disponível em: http://opower.com/. Data de acesso: 26-06-2012

[15]. Soluções de serviços públicos da Silver Spring Networks - A Plataforma de Energia Inteligente. Disponível em: http://www.silverspringnet.com/pt/solutions/. Data de acesso: 26-06-2012

[16]. Residential Energy Efficiency: Save on Air Conditioning, Heating, Electric Heat, and More. Disponível em: http://www.powerhousedynamics.com/residential-energy-efficiency/. Data de acesso: 26-06-2012

[17]. Tendril » Tendril Connect™ Energy Platform » Connect. Disponível em: http://www.tendrilinc.com/platform/connect/. Data de acesso: 26-06-2012

[18]. LEM Website - Wi-LEM. Disponível em: http://www.lem.com/hq/en/content/view/276/215/. Data de acesso: 26-06-2012

[19]. EnergyHub. Disponível em: http://www.energyhub.com/. Data de acesso: 26-06-2012

[20]. IsGreen. *Greenlamp Command & Control C&C*. Disponível em: http://www.isgreen.pt. Data de acesso: 15-09-2011

[21]. Software de gestão de energia Energy-Brain. *Energy-Brain software de contagem e gestão de energia*. em: http://www.qenergia.pt/129/software-de-gestao-de-energia-energy-brain.htm. Data de acesso: 15-09-2011

[22]. Optimal. *Optimal Monitoring: Energy Monitoring Demonstration*. Disponível em: http://www.optimalmonitoring.com/What-Is-Optimal.html. Data de acesso: 15-09-2011

[23]. Hughes Energy System. *eSight – advanced, fully web enabled energy monitoring, targeting and analysis so*. Disponível em: http://www.hughes-energy.com/#/esight/4549668305. Data de acesso: 15-09-2011

[24]. MeterRead™SALE (Go Green, Save Money, Save Earth) Version: 1.2.1 Review | iPhone and iPad Finance App | Macworld. Disponível em: http://www.macworld.com/appguide/app.html?id=71548&expand=false. Data de acesso: 25-06-2012

[25]. Zerogate. Disponível em: www.zerogate.com. Data de acesso: 25-06-2012

[26]. App Store - Energy UFO. Disponível em: http://itunes.apple.com/app/energy-ufo/id303338274?mt=8. Data de acesso: 25-06-2012

[27]. Palo Alto CA Energy Management Systems - Visible Energy Inc Energy Savings Company. Disponível em: http://www.visiblenergy.com/. Data de acesso: 25-06-2012

[28]. App Store - Meter Readings. Disponível em: http://itunes.apple.com/pt/app/meter-readings/id320551309?mt=8. Data de acesso: 25-06-2012

[29]. Graham Haley. Disponível em: http://grahamhaley.co.uk/. Data de acesso: 25-06-2012

[30]. jCharts. Disponível em: http://jcharts.sourceforge.net/index.html. Data de acesso: 25-06-2012

[31]. java charts and graphs – EasyCharts. Disponível em: http://www.objectplanet.com/easycharts/. Data de acesso: 25-06-2012

[32]. Objectplanet - Company. Disponível em: http://www.objectplanet.com/company.html. Data de acesso: 25-06-2012

[33]. JFreeChart. Disponível em: http://www.jfree.org/jfreechart/. Data de acesso: 25-06-2012

[34]. Java Chart - JenSoft API. Disponível em: http://www.jensoft.org/jensoft/Home. Data de acesso: 25-06-2012

[35]. Three D Graphics: Corporate Graphics Solutions, Consumer Graphics Software, Financial Analysis Tools. Disponível em: http://www.threedgraphics.com/tdg/products/tools/ioschart/. Data de acesso: 25-06-2012

[36]. iPhone, iPad Line Chart for iPhone Objective-C. Disponível em: http://www.keepedge.com/iphone_charts/line_chart.html. Data de acesso: 25-06-2012

[37]. core-plot - Cocoa plotting framework for OS X and iOS. Disponível em: http://code.google.com/p/core-plot/. Data de acesso: 25-06-2012

[38]. Google Chart Tools - Google Developers. Disponível em: https://developers.google.com/chart/. Data de acesso: 25-06-2012

[39]. Goodies and freebies | FusionCharts. Disponível em: http://www.fusioncharts.com/goodies/fusioncharts-free/. Data de acesso: 25-06-2012

[40]. amCharts: JavaScript/HTML5 charts. Disponível em: http://www.amcharts.com/. Data de acesso: 25-06-2012

[41]. File: Scrum process. Disponível em: http://en.wikipedia.org/wiki/File:Scrum_process.svg. Data de acesso: 21-06-2012

[42]. Scrum Alliance – What is Scrum?. Disponível em: http://www.scrumalliance.org/learn_about_scrum. Data de acesso: 21-06-2012.

[43]. TurtoiseSVN. Disponível em: http://tortoisesvn.net/. Data de acesso: 24-06-2012

[44]. Home :: Bugzilla: bugzilla.org. Disponível em: http://www.bugzilla.org/. Data de acesso: 24-06-2012

[45]. FERREIRA, F. *Sistema de monitorização de consumos para o DEEC: Definição da estrutura física de recolha de dados e de interface com os utentes*. 2011. 122 p. Dissertação (Mestrado Integrado em Engenharia Eletrotécnica e de Computadores) – Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (DEEC-FCTUC), Coimbra, 2011. (PDF).

[46]. FERREIRA, F. *Documento de apoio à definição do interface com os utentes*. 2011. 46p. Dissertação (Mestrado Integrado em Engenharia Eletrotécnica e de Computadores) – Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (DEEC-FCTUC), Coimbra, 2011. (PDF).

[47]. IEEE Std 830-1948. *Guide to Software Requirements Specifications*. IEEE Standards Collection: Software Engineering. Piscatway, NJ: The Institute of Electrical and Electronics Engineers, 193. (PDF).

[48]. IEEE Std 830-1998. *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Standards Collection: Software Engineering. Piscatway, NJ: The Institute of Electrical and Electronics Engineers, 1998. (PDF).

[49]. IEEE Std 1371-2000. *IEEE Recommended Practice for Arquitectural Description of Software-Intensive Systems*. 2000. (PDF).

[50]. HILLIARD, R. IEEE Std 1471-2000. *IEEE-Std-1471-2000. Recommended Practice for Arquitectural Description of Software-Intensitve Systems*. 2000. (PDF).

[51]. UML, BPMN and Database Tool for Software Development. Disponível em: http://www.visual-paradigm.com/. Data de acesso: 18-11-2011.

[52]. iOS Dev Center. *Apple Developer*. Disponível em: http://developer.apple.com. Data de acesso: 05-01-2012

[53]. Design Patterns | Object Oriented Design. Disponível em: http://www.oodesign.com/. Data de acesso: 01-01-2012

[54]. Oracle Technology Network. *How to Write Doc Comments for the Javadoc Tool*. Disponível em: http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html. Data de acesso: 01-12-2011

[55]. docs.Oracle. *javadoc-The Java API Documentation Generator*. Disponível em: http://docs.oracle.com/javase/1.3/docs/tooldocs/win32/javadoc.html#javadoctags. Data de acesso: 01-12-2011

[56]. CentoOS. Disponível em: http://www.centos.org/. Data de acesso: 01-01-2012

[57]. WinSCP :: Free SFTP and FTP client for Windows. Disponível em: http://winscp.net/eng/index.php. Data de acesso: 01-01-2012

[58]. Download PuTTY - a free SSH and telnet client for Windows. Disponível em: http://www.putty.org/. Data de acesso: 01-01-2012

[59]. High Performance PHP Framework for Web Development - Symfony. Disponível em: http://symfony.com/. Data de acesso: 01-06-2012

[60]. Java SE Application Design with MVC. Disponível em: http://www.oracle.com/technetwork/articles/javase/index-142890.html. Data de acesso: 01-01-2012