

Mestrado em Engenharia Informática
Estágio
Relatório Final

**Detecção de padrões de consumo de energia
com CBR**

João Alexandre dos Santos Saraiva

jass@student.dei.uc.pt

Orientadores

DEI | FCTUC

Alexandre Pinto

ISA | Intelligent Sensing Anywhere

António Damasceno

01 de Julho de 2013



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



Intelligent Sensing
Anywhere

Agradecimentos

Quero agradecer a todas as pessoas que me apoiaram durante o decorrer deste estágio e que me ajudaram a ultrapassar os obstáculos que foram aparecendo ao longo do tempo. Em especial à minha família e amigos que sempre me apoiaram em todos os momentos.

Agradeço também aos meus orientadores do estágio pela sua disponibilidade e ajuda prestada.

Resumo

Numa época de aumento de custos para as famílias portuguesas, a monitorização da energia em ambientes domésticos pode ser uma forma de poupar nos gastos de electricidade, reduzindo assim os custos por mês.

Pretende-se fazer uma identificação, a partir de uma série temporal de consumo energético, dos equipamentos eléctricos utilizados durante o período de tempo analisado, identificando quando foram ligados e desligados.

Através deste estágio pretende-se provar que é possível identificar padrões de consumo de energia com *case-based reasoning*. A aplicação produzida ao longo do estágio não será uma aplicação comercial, sendo apenas uma prova de conceito. Após o estágio e uma análise de mercado da parte da ISA, a aplicação pode dar origem a uma aplicação comercial.

Palavras-chave: Case-based reasoning, *Clology*, Extração de características, Reconhecimento de padrões.

Índice

Agradecimentos.....	3
Resumo	5
Capítulo 1 Introdução	1
Capítulo 2 Trabalho relacionado e Estado da Arte	5
Capítulo 3 Requisitos.....	13
Capítulo 4 Arquitectura	21
Capítulo 5 Análise de dados efectuada	27
Capítulo 6 Planeamento	35
Capítulo 7 Desenvolvimento	41
Capítulo 8 Testes.....	45
Capítulo 9 Conclusões e Trabalho futuro	47
Bibliografia.....	49

Capítulo 1

Introdução

1.1 Contextualização

O presente estágio surge devido à necessidade da empresa ISA (*Intelligent Sensing Anywhere, S.A.*) de detectar padrões de consumo de energia eléctrica em ambientes domésticos para os seus clientes.

O mercado-alvo da aplicação pretendida são as pessoas que pretendam ver a sua factura energética doméstica diminuir para poupar dinheiro ou porque pensam que estão a fazer uma má utilização dos seus equipamentos eléctricos. Desta forma, os segmentos da população mais afectados vão ser a classe média e a classe baixa, visto que são o tipo de pessoas mais interessadas em poupar as suas economias e reduzir os seus gastos o mais possível.

Este estágio surge também devido ao facto do DEI (Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra) convidar Empresas e Instituições a proporem trabalhos de dissertação/estágio para serem realizados pelos alunos de Mestrado em Engenharia Informática.

A ISA é uma empresa de base tecnológica, que oferece e implementa soluções e serviços inovadores de telemetria e gestão remota. Desenvolve sensores com tecnologia inteligente que transmitem informação para plataformas de gestão.

Especialista e líder em soluções de telemetria e gestão remota nas áreas da energia, gás, petróleo, saúde e ambiente, é reconhecida internacionalmente e está hoje presente em mais de 20 países dos cinco continentes do mundo.

O cariz fortemente exportador, bem como a qualidade técnica e inovadora das suas soluções, tem colocado a empresa numa posição competitiva de relevo, tanto a nível nacional como internacional, tendo já o BES, EDP, BP, Galp, PT, EPAL, Águas do Mondego, Águas de Coimbra, Centro Hospitalar de Coimbra, ANA – Aeroportos, PRIMAGAZ, BUTAGAZ, DIGAL, SANASA ou a Universidade de Aveiro, como alguns dos principais clientes.

O seu percurso começou há mais de 20 anos, com a aposta na telemetria de gás, área onde é pioneira e líder mundial.

Sendo uma empresa 100% portuguesa em constante crescimento, com 120 colaboradores que trabalham diariamente com tecnologia de ponta para levar as melhores soluções e os serviços mais adequados a instituições, empresas e residências de todo o mundo.

1.2 Identificação do problema

Numa época de aumento de custos para as famílias portuguesas, a monitorização da energia em ambientes domésticos pode ser uma forma de poupar nos gastos de electricidade, reduzindo assim os custos por mês.

Pretende-se fazer uma identificação, a partir de uma série temporal de consumo energético, dos equipamentos eléctricos utilizados durante o período de tempo analisado, identificando quando foram ligados e desligados.

Através deste estágio pretende-se provar que é possível identificar padrões de consumo de energia com *case-based reasoning*. A aplicação produzida ao longo do estágio não será uma aplicação comercial. Após o estágio e uma análise de mercado da parte da ISA, a aplicação pode dar origem a uma aplicação comercial.

1.3 Motivação

Ao conseguir efectuar uma correcta identificação dos padrões de consumo energético dos vários equipamentos eléctricos e detectar quando estes estão a ser utilizados, pode-se utilizar esta informação para alcançar diferentes objectivos.

Uma das aplicações possíveis é a identificação do comportamento dos equipamentos eléctricos nas casas de clientes da ISA, possibilitando posteriormente um melhor uso dos mesmos e poupanças de energia aos clientes, reduzindo assim a factura de electricidade no final do mês.

A identificação dos padrões de consumo energético dos vários equipamentos eléctricos pode também ser utilizada para detectar alguma avaria que ocorra num equipamento que está sempre ligado, como é o caso de arcas congeladoras. Neste caso pode evitar que a comida armazenada nestas se estrague, poupando assim dinheiro ao utilizador.

A própria ISA beneficia também com este produto, visto que informatiza uma tarefa que actualmente é feita por humanos, tornando assim este sistema independente e libertando estas pessoas para outras tarefas que lhes possam ser alocadas.

Ao diminuir o consumo de electricidade dos clientes, vai também beneficiar a população em geral e o ambiente, visto que diminui por consequência a emissão de CO₂ (Dióxido de Carbono).

1.4 Características da solução pretendida

O *Cloogy*, é a solução actual da ISA para monitorização da energia em ambientes domésticos, através de um *clamp* colocado à entrada do quadro eléctrico e da monitorização e actuação em tomadas individuais usando tomadas inteligentes.

A detecção de padrões de consumo de energia com *case-based reasoning* (CBR) pretende identificar comportamentos de equipamentos eléctricos.

A solução que se pretende agora desenvolver tirará partido do *Cloogy* para obter dados de base sobre os quais aplicará algoritmos de CBR para conseguir identificar e distinguir quais os equipamentos eléctricos em funcionamento ao longo do tempo, bem como a altura em que estes são ligados e desligados.

Pretende-se utilizar os dados obtidos pelas tomadas individuais para extracção de características dos padrões de consumo de energia de cada equipamento eléctrico e para treino da plataforma.

Pretende-se que a solução a desenvolver seja capaz de desagregar os dados da série temporal de consumo energético obtida do *clamp* colocado à entrada do quadro eléctrico para identificar quais os equipamentos eléctricos em utilização nessa mesma série temporal.

Inicialmente o foco deste estágio era identificar correctamente os padrões de consumo de energia do frigorífico, máquina de lavar a louça e máquina de lavar a roupa, isto porque estes são os electrodomésticos normalmente mais utilizados e com maior consumo de energia.

Posteriormente este foco foi reduzido para a identificação dos padrões de consumo de energia da máquina de lavar a louça e máquina de lavar a roupa, devido ao facto do frigorífico não ter ciclos de funcionamento com tempo regular se existir uma anomalia, ou se, por exemplo, a porta do frigorífico estiver aberta durante muito tempo. Ao contrário do frigorífico, a máquina de lavar a roupa e a máquina de lavar a louça têm comportamentos regulares ao longo do tempo, sendo que cada programa demora um determinado tempo, sendo este fixo.

Os objectivos deste estágio são a implementação do ciclo de *case-based reasoning* e não dos algoritmos de aprendizagem e reconhecimento de padrões, que serão invocados pelo ciclo CBR.

1.5 Planeamento preliminar

A empresa ISA tem as certificações ISO 9001 de gestão de qualidade, NP 4457 relativa à gestão da investigação, desenvolvimento e inovação e ainda também a certificação CMMI-DEV de desenvolvimento de *software* de nível de maturidade 2. As directrizes da ISA obrigam o aluno a cumprir o processo de

concepção e desenvolvimento da ISA ao longo do estágio. Ao cumprir este processo, é garantido que são cumpridas as normas ISO 9001, NP 4457 e CMMI-DEV de nível 2 de maturidade, visto que os auditores da empresa aprovaram o processo de concepção e desenvolvimento, estando este em conformidade com as normas.

Capítulo 2

Trabalho relacionado e Estado da Arte

2.1 Trabalho relacionado

2.1.1 Case-based reasoning

Case-based reasoning (CBR) é uma metodologia que surgiu a partir de uma investigação em ciência cognitiva efectuada por Roger Schank [1] [2] [3] e alguns alunos, dos quais era docente na Universidade de Yale, USA. As suas origens foram estimuladas pelo desejo de entender como é que o ser Humano se lembra de informação que contém no seu cérebro e utiliza esta para resolver problemas. Foi confirmado que as pessoas resolvem problemas frequentemente, lembrando-se de como resolveram problemas semelhantes anteriormente.

A definição de CBR foi inventada por Riesbeck e Schank [4]: “A *case-based reasoner solves problems by using or adapting solutions to old problems*”.

O CBR é normalmente definido pelo seu ciclo, que contém quatro actividades:

- 1- *Retrieve* – recuperar casos semelhantes ao da descrição do problema;
- 2- *Reuse* – reutilizar uma solução sugerida por um caso semelhante;
- 3- *Revise* – rever ou adaptar essa solução para melhor atender ao problema, se necessário;
- 4- *Retain* – manter a nova solução depois de confirmada ou validada.

De seguida é apresentada uma imagem que ilustra o ciclo da metodologia *case-based reasoning*, descrito anteriormente.

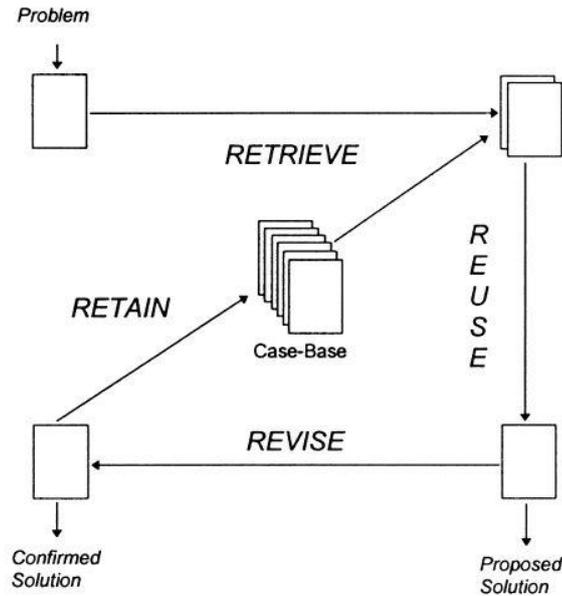


Fig. 1 – Ciclo CBR [5]

Existem diferentes tecnologias que podem ser utilizadas em *case-based reasoning*. Isto é possível, visto que o CBR é uma metodologia e não uma tecnologia e desta forma, pode-se implementar uma solução CBR de várias maneiras, recorrendo a diferentes tecnologias e atingindo o mesmo objectivo.

2.1.1.1 Vizinho mais próximo

As técnicas de vizinho mais próximo [5] são as mais utilizadas em *case-based reasoning*. Para cada atributo é determinada a semelhança de cada caso com o problema em questão. Podendo depois multiplicar esta medida por um factor ponderado. Posteriormente, a soma da semelhança de todos os atributos é calculada para obter uma medida da semelhança do caso guardado com o caso alvo. A seguinte equação representa esta operação:

$$\text{Semelhança}(T, S) = \sum_{i=1}^n f(T_i, S_i) \times w_i \quad (2.1)$$

em que T (*target* do Inglês) é o caso alvo, S (*source* do Inglês) é o caso guardado, n é o número de atributos de cada caso, i é cada atributo de 1 até n, f é uma função de semelhança para o atributo i para os casos T e S, e w é o peso que esse atributo tem.

O objectivo da técnica de vizinho mais próximo é calcular a distância entre o caso alvo e todos os casos guardados, mantendo o caso guardado com

distância menor até ao momento, retornando no final o caso guardado mais semelhante ao caso alvo. Existem várias variantes da técnica de vizinho mais próximo, como é o caso do k-vizinhos mais próximos, em que são identificados os k vizinhos mais semelhantes ao caso alvo, sendo o caso alvo classificado de acordo com os k vizinhos mais próximos.

2.1.1.2 Cálculo de semelhança

Existem várias funções que podem ser utilizadas para calcular a semelhança entre dois casos.

O cálculo da semelhança entre os casos tem como objectivo complementar algumas das técnicas referidas anteriormente, como é o caso da técnica de vizinho mais próximo, em que é necessário uma função de semelhança.

Esta função de semelhança pode ser o cálculo de uma distância euclidiana simples entre dois pontos, uma distância euclidiana com pesos ou uma outra função, dependendo do contexto do trabalho em questão [6].

A distância euclidiana simples entre dois pontos é dada pela seguinte função:

$$d_{a,b} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (2.2)$$

em que a e b representam dois pontos definidos pelas coordenadas cartesianas (x_a, y_a) e (x_b, y_b) , respectivamente.

A distância euclidiana com pesos deriva da distância euclidiana simples, apresentada em (2.2). Desta forma, temos que:

$$\begin{aligned} d_{a,b} &= \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \\ &= \sqrt{\left(\frac{x_a - \bar{x}}{s_x} - \frac{x_b - \bar{x}}{s_x}\right)^2 + \left(\frac{y_a - \bar{y}}{s_y} - \frac{y_b - \bar{y}}{s_y}\right)^2} \\ &= \sqrt{\left(\frac{x_a}{s_x} - \frac{x_b}{s_x}\right)^2 + \left(\frac{y_a}{s_y} - \frac{y_b}{s_y}\right)^2} \\ &= \sqrt{\frac{1}{s_x^2}(x_a - x_b) + \frac{1}{s_y^2}(y_a - y_b)} \end{aligned} \quad (2.3)$$

sendo que $\frac{1}{s_x^2}$ e $\frac{1}{s_y^2}$ são os pesos, que são o inverso da variância.

A função de semelhança pode ainda ser uma outra função, dependendo do contexto do trabalho. No caso particular deste estágio, em que um padrão de

consumo energético de um equipamento eléctrico é representado por um vector com as características desse padrão, a função de semelhança é a seguinte:

$$Sim(avg, sdv, value) = density\left(\frac{|value - avg|}{sdv}\right) \quad (2.4)$$

em que *avg* é a média no ponto *x*, *sdv* o desvio padrão no ponto, *value* é o valor a testar e *density(x)* retorna a função densidade de probabilidade de uma distribuição com média 0 e desvio padrão 1, no ponto *x*. Esta função de semelhança é utilizada na abordagem não paramétrica, descrita na subsecção 5.2.3.

2.1.2 Extracção de características

Como na aplicação deste estágio vai ser utilizado CBR para a classificação de casos, tem de existir um repositório com os casos anteriores, bem como uma forma de comparar os novos casos com os casos armazenados.

Para fazer a comparação entre os casos é necessário saber quais as características que são importantes para cada caso. São estas características de cada série temporal de consumo energético que identificam o caso presente na série, sendo fundamental descobrir quais as características que identificam cada caso e os métodos para extrair as mesmas.

A extracção de características tem como objectivo caracterizar a série temporal de consumo energético que vai ser analisada para identificação de qual o equipamento ligado no respectivo espaço de tempo desta mesma série.

Esta caracterização pode ser feita utilizando vários valores estatísticos, sendo que quanto mais valores forem utilizados, maior e melhor será a caracterização efectuada, desde que estes valores sejam relevantes.

No caso particular deste estágio, vão ser realizadas diferentes caracterizações para os diferentes equipamentos eléctricos. Por exemplo, a caracterização de uma máquina de lavar a roupa vai ser diferente da caracterização feita para a máquina de lavar a louça.

Isto deve-se ao facto de os equipamentos eléctricos terem consumos de energia diferentes e os padrões temporais de consumo de energia gerados por estes consumos serem diferentes.

Por exemplo, a máquina de lavar a louça tem normalmente um ciclo de pré-lavagem, seguido de um ciclo de lavagem principal em que a água é aquecida e misturada com o detergente, ciclos intermediários de enxaguamento e por fim o ciclo de secagem.

No caso da máquina de lavar a roupa o comportamento é diferente, visto que primeiro é aquecida a água, consumindo uma determinada energia, posteriormente é efectuada a lavagem e no final é torcida a roupa.

Desta forma, temos neste exemplo, dois comportamentos diferentes nestes electrodomésticos específicos, originando diferenças nos gráficos de consumo de energia.

A extracção de características e o reconhecimento de padrões são feitos por algoritmos desenvolvidos pela ISA, não sendo da autoria e responsabilidade do mestrando.

O ciclo CBR da aplicação, da responsabilidade e autoria do mestrando, será suficientemente flexível para poder fazer uso de variados algoritmos alternativos de extracção de características e de reconhecimento de padrões.

2.1.3 Reconhecimento de padrões

Um padrão pode ser considerado como um conjunto de características semelhantes. Também pode ser a descrição de um problema recorrente para o qual existe uma solução que pode ser reutilizada várias vezes em situações diferentes.

Em ambos os casos, são questões que envolvem uma série de características que se repetem e podem ser devidamente identificadas. Desta forma, estas passam a compor um comportamento rotineiro cuja resolução pode estar disponível de forma constante.

Existem várias técnicas de reconhecimento de padrões, tais como K-vizinhos mais próximos [7], Redes neuronais [8] ou Máquinas de vectores de suporte [8], que não vão ser analisadas, devido ao facto de se pretender provar, através do presente estágio, que é possível identificar padrões de consumo de energia de equipamentos eléctricos, em ambientes domésticos, fazendo uso de *case-based reasoning*.

Uma vez que é este o objectivo do estágio, a aplicação desenvolvida ao longo do mesmo não será uma aplicação comercializada.

Posteriormente, a empresa ISA pode estudar uma forma de aplicar uma solução comercial de *case-based reasoning*, desenvolvendo uma aplicação comercial, utilizando o trabalho realizado neste estágio.

O reconhecimento de padrões será então efectuada utilizando CBR e caracterização estatística.

2.1.4 Conclusão

Para a solução a ser implementada neste estágio, vai ser utilizado CBR e caracterização estatística para o reconhecimento de padrões. Posteriormente, dependendo dos resultados obtidos podem ser introduzidas outras técnicas de reconhecimento de padrões no CBR, como é o caso de redes neuronais, por exemplo.

2.2 Estado da arte

2.2.1 AppliSense

Este é um sistema que utiliza um único medidor inteligente de electricidade, colocado à entrada do quadro eléctrico, que tem uma interface de comunicação integrada a um *gateway*, responsável pela contínua obtenção de dados e armazenamento através do medidor de electricidade. É também responsável pelo tratamento de pedidos da interface do utilizador.

A interface do utilizador é uma aplicação de *smartphone*, que apresenta o seu consumo de electricidade em tempo real.

A comunicação utilizada entre o *gateway* e o *smartphone* é efectuada através de *REST*.

O consumo de electricidade altera-se durante o tempo, conforme a utilização de vários equipamentos na casa. Os instantes em que se liga e desliga um equipamento eléctrico têm um nível de consumo de electricidade idêntico.

O AppliSense [9] começa por identificar os locais em que ocorrem mudanças significativas entre dois níveis de energia.

Uma vez detectados estes locais, calcula a diferença entre os níveis de energia consecutivos detectados, classificando-os como um provável evento de ligação de um equipamento eléctrico.

Por último, compara cada uma das diferenças descritas anteriormente com um conjunto de diferenças guardadas na sua base de dados, que caracterizam cada aparelho eléctrico, classificando assim a ocorrência como o equipamento que contém as mesmas características.

Esta aplicação identifica os diferentes equipamentos eléctricos em utilização através da comparação de dados com uma base de dados.

2.2.2 SUMMER

O SUMMER [10] (*Salford University Monitoring and Management Energy Research*) é um projecto da Universidade de Salford que tem como objectivo recolher dados de quem gasta energia e como o faz, em ambientes domésticos.

Para conseguir atingir os objectivos deste projecto são recolhidos dados de interacções entre as pessoas e o consumo de energia, como é o caso de ligar e desligar equipamentos eléctricos.

São também recolhidos dados referentes à localização das pessoas em casa, para perceber quem está a beneficiar do equipamento eléctrico (quem está ver televisão ou utilizar o computador) e padrões de consumo de energia, de modo a que as interacções sejam atribuídas às pessoas correctas.

De modo a recolher os dados referidos anteriormente, são utilizados equipamentos de controlo de consumo de energia, bem como sensores para localizar a presença das pessoas em casa.

Este projecto divide-se em quatro etapas.

A primeira consiste em testar a tecnologia para detectar a interacção das pessoas com os equipamentos que consomem energia em casa. Estes testes são efectuados numa casa tradicional vitoriana anterior a 1920, reconstruída fazendo uso de materiais recicláveis, numa câmara de ambiente controlável, na Universidade de Salford.

Após os testes efectuados na casa anteriormente descrita, vão ser efectuados novos testes, desta vez em *Living Labs* no Reino Unido. *Living Labs* são comunidades voluntárias em Manchester, Birmingham e Bristol que testam tecnologia, dão o seu feedback e partilham as suas experiências com outros participantes.

Posteriormente à recolha de dados, os resultados são analisados para descrever padrões de comportamento. Esta informação é bastante valiosa para todas as partes envolvidas na redução do consumo de energia doméstica, concedendo uma visão de como as pessoas realmente fazem uso da energia nas suas casas.

2.2.3 Entracker

Este [11] é um projecto realizado na Universidade de Brown, que tem como objectivo fornecer uma estimativa em tempo real, da potência por equipamento eléctrico num ambiente doméstico.

Para isto, é utilizado um medidor de energia que grava o consumo total da energia de toda a habitação e um dispositivo que determina o estado binário (ligado ou desligado) dos dispositivos na mesma.

Cada equipamento eléctrico comporta-se de uma determinada forma quando é ligado, enquanto está ligado e quando é desligado, alterando assim o consumo de energia nestas três fases distintas.

O que é feito inicialmente é ligar cada equipamento eléctrico várias vezes, obtendo assim o comportamento de cada um, classificando-o manualmente para construir um conjunto de casos conhecidos. Para esta classificação são utilizadas máquinas de vectores de suporte (SVM).

A classificação final do equipamento que foi ligado é feita através de uma combinação do uso de SVM e da magnitude registada quando se liga o equipamento. Esta magnitude é caracterizada pelos autores do projecto como “Euclidean transient”, visto que a técnica utilizada para a obtenção desta é a Distância Euclidiana.

Cada uma das técnicas descritas anteriormente, retornam uma classificação relativa aos casos guardados no primeiro passo. Estas duas classificações individuais são então combinadas numa só, permitindo determinar qual o equipamento que foi ligado.

A técnica de SVM utilizada é uma *binary SVM classification* que classifica para cada equipamento eléctrico os dados da seguinte forma: cria dois grupos para cada equipamento, classificando-os como “equipamento X” e “não equipamento X”, sendo que todos os casos conhecidos do equipamento X fazem parte do primeiro e todos os outros fazem parte do segundo grupo.

Tal como o projecto anterior (SUMMER) [10], este é efectuado num laboratório de uma Universidade, ou seja, é criado um ambiente doméstico artificial neste espaço para a realização destes projectos.

2.2.4 Conclusão

Nenhuma das soluções referidas anteriormente utiliza a metodologia CBR para detectar padrões de consumo de energia eléctrica. É neste sentido que o presente estágio dá um contributo novo, explorando o uso de CBR para esta finalidade.

Capítulo 3

Requisitos

3.1 Metodologia

A metodologia que vai ser utilizada neste estágio para definição dos requisitos que o compõem é chamada de MoSCoW, visto que esta é utilizada pela ISA na realização dos seus projectos.

MoSCoW é uma técnica de priorização usada em desenvolvimento de *software* para chegar a uma consonância com os *stakeholders* sobre a importância de cada requisito.

A metodologia MoSCoW [12] de análise de requisitos, que vai ser utilizada neste estágio, divide os requisitos em quatro categorias diferentes: **Must**, **Should**, **Could** e **Won't**. A descrição das categorias é a seguinte:

- **Must** – Descreve um requisito que tem de ser concretizado na solução final, para que esta seja considerada um sucesso;
- **Should** – Representa um item de alta-prioridade que deve ser incluído na solução, caso seja possível. Este é normalmente um tipo de requisito essencial, mas que pode ser satisfeito por outros meios se for estritamente necessário;
- **Could** – Tipo de um requisito que é considerado desejável mas não necessário. Estes requisitos são incluídos se existir tempo e recursos que o permitam;
- **Won't** – Representa um requisito que os *stakeholders* concordaram que não seria implementado na solução final, mas pode ser considerado posteriormente.

Existem várias ferramentas que podem ser utilizadas para a criação de requisitos, sendo que neste estágio vai ser utilizada a ferramenta *Enterprise Architect*, visto que esta é usada para todos os projectos da empresa para este efeito, bem como para gerar a Arquitectura dos mesmos projectos.

Na seguinte secção 3.2 são apresentados os requisitos identificados para este trabalho.

3.2 Análise de requisitos

Como já foi referido anteriormente neste relatório, a aplicação resultante do presente estágio não vai ser uma aplicação comercial.

Desta forma, em relação a atributos de qualidade da aplicação, esta não tem que ter um certo tipo de eficiência, tempo de resposta, segurança ou usabilidade, visto que não vai ser uma aplicação comercial.

O atributo de qualidade que a aplicação deve respeitar é fiabilidade, ou seja, a solução encontrada deve conseguir identificar os equipamentos e distingui-los.

Não deve haver, por exemplo, um caso em que dois equipamentos eléctricos distintos sejam identificados como sendo o mesmo, por terem um comportamento semelhante, nem classificar dois equipamentos eléctricos como sendo distintos, quando na realidade são o mesmo, para garantir a fiabilidade referida anteriormente.

De seguida, é apresentada uma imagem que ilustra o diagrama dos casos de uso referentes ao estágio identificados, seguindo-se uma descrição dos mesmos posteriormente.

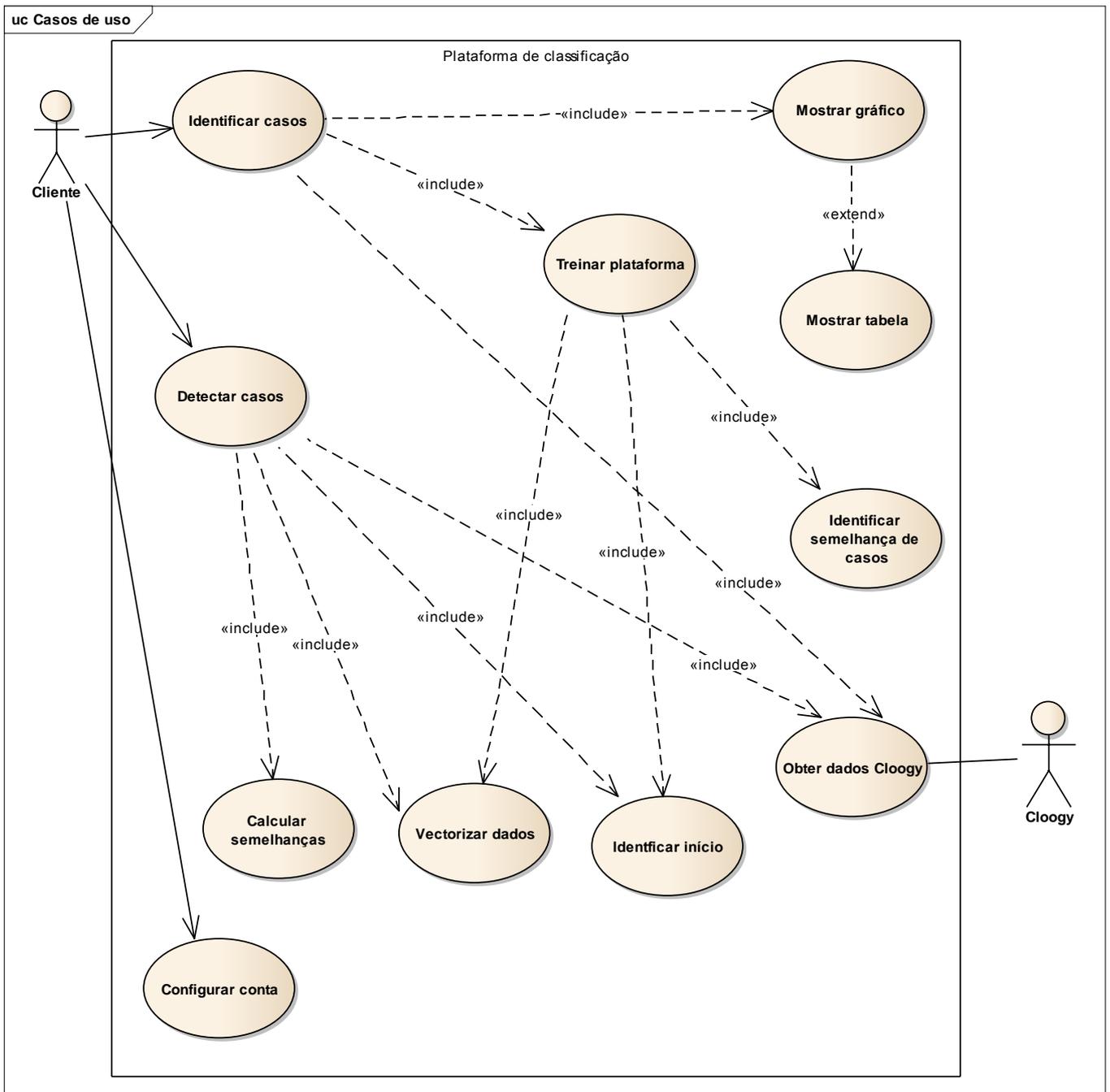


Fig. 2 – Diagrama de casos de uso

3.3 Descrição dos casos de uso

3.3.1 Caso de uso 1: Identificar casos

Este caso de uso pretende ilustrar a situação em que são identificados os casos que vão ser guardados para futuras classificações. Ou seja, estes casos vão ser os casos típicos de comportamento dos vários equipamentos eléctricos para treinar o sistema de *case-based reasoning*.

Desta forma, posteriormente quando for pretendido identificar a que equipamento eléctrico corresponde um certo padrão de consumo de energia, esse padrão vai ser comparado com estes casos guardados no sistema.

Os casos típicos de comportamento dos vários equipamentos eléctricos são guardados em pastas, sendo que cada pasta contém os casos de um equipamento eléctrico. É importante que a pasta correspondente a um determinado equipamento eléctrico não tenha casos de outro equipamento, porque se isto acontecer vai originar classificações erradas de equipamentos eléctricos.

Este é um requisito **Must**, visto que sem haver uma identificação de casos é impossível haver uma classificação posterior de casos.

3.3.2 Caso de uso 2: Mostrar gráfico

Neste caso de uso, a aplicação apresenta um gráfico ilustrando o consumo energético efectuado pelo equipamento eléctrico, guardado na pasta referente ao equipamento eléctrico respectivo, referido no caso de uso anterior, exibindo assim o comportamento desse mesmo equipamento eléctrico neste caso específico, ou seja, o padrão de consumo de energia deste equipamento.

Trata-se de um requisito **Should**, dando a possibilidade ao utilizador de ter uma caracterização visual do comportamento do equipamento eléctrico em questão.

3.3.3 Caso de uso 3: Mostrar tabela

Em vez de mostrar um gráfico para ilustrar o consumo energético do equipamento eléctrico identificado anteriormente, pode ser apresentada uma tabela, contendo o consumo de energia detalhado.

Desta forma, o utilizador consegue observar a evolução do consumo energético do caso em questão mais detalhadamente, visto que tem ao seu dispor os valores de consumo de energia ao longo do tempo deste caso organizados

numa tabela, ao contrário de um gráfico, em que apenas são representados estes valores.

A granularidade dos dados e colunas da tabela serão detalhados quando o caso de uso for implementado.

Visto que o caso de uso anterior, referente à apresentação de um gráfico ilustrativo do consumo de energia do caso identificado é identificado como *Should*, este caso de uso é um **Must**, porque ambos os casos são referentes à apresentação dos mesmos dados, embora de formas diferentes.

3.3.4 Caso de uso 4: Treinar plataforma

Este caso de uso pretende ilustrar a situação em que são guardados na plataforma de *case-based reasoning*, os casos identificados pelo utilizador, com a respectiva caracterização dos mesmos.

Estes casos vão ser posteriormente usados na identificação do equipamento eléctrico que tem um determinado padrão de consumo de energia que pretende ser identificado, através da comparação com estes e da semelhança entre eles.

Este processo consiste em utilizar os casos de uso: Identificar início, Vectorizar dados e Identificar semelhança de casos.

Este é um requisito **Must**, visto que tem de existir um treino da plataforma, para que posteriormente através da comparação de casos guardados com os casos em estudo, possam ser identificados os equipamentos eléctricos aos quais pertencem os casos em estudo.

3.3.5 Caso de uso 5: Identificar semelhança de casos

Uma vez treinada a plataforma, os casos têm de ter uma ligação entre si, caso sejam do mesmo equipamento eléctrico.

Faz sentido que a plataforma contenha vários casos guardados do mesmo equipamento eléctrico, visto que o mesmo equipamento eléctrico não tem um comportamento e conseqüente consumo de energia igual sempre que é ligado.

Desta forma, este caso de uso pretende expressar a identificação de semelhança entre casos do mesmo equipamento eléctrico.

Este requisito é classificado como **Must**, porque tem de haver uma semelhança entre os vários casos do mesmo equipamento eléctrico guardados na plataforma, agrupando-os conforme esta característica.

3.3.6 Caso de uso 6: Obter dados Cloogy

Este caso de uso pretende caracterizar a situação em que são obtidos dados do *Cloogy*.

O *Cloogy* é a solução da ISA para monitorização de energia em ambiente doméstico, permitindo a monitorização do consumo geral de toda a habitação.

Desta forma, é necessário que a aplicação desenvolvida neste estágio tenha interacção com o *Cloogy*. Esta interacção é efectuada para recolher informação sobre o consumo de energia efectuado na habitação ao longo do tempo.

Estes dados vão servir para a identificação de casos de consumo de energia dos vários equipamentos eléctricos, referida no caso de uso 1, e posteriormente para a detecção de casos através da comparação com os anteriores.

Visto que a obtenção destes dados é necessária para a identificação e detecção de casos, este é também um requisito **Must**.

3.3.7 Caso de uso 7: Identificar início

Para ser possível identificar e detectar um caso, é necessário que seja detectado o início desse caso, sendo que o presente caso de uso pretende descrever essa mesma situação.

Por exemplo, numa série temporal de consumo energético referente ao consumo eléctrico verificado numa determinada tomada eléctrica à qual está ligada a máquina de lavar a roupa, podem existir vários eventos de funcionamento deste electrodoméstico.

Consequentemente é necessário identificar o início de cada caso em que se verifica o início do funcionamento deste electrodoméstico, para poder isolar os casos, caracterizá-los e guarda-los ou então detectar a que se referem estes casos, dependendo se está a ser identificado um caso ou detectado um caso, respectivamente.

Este é um requisito **Must**, visto que para poder ser identificado ou detectado um caso de consumo de energia de um equipamento eléctrico específico, é necessário que seja detectado o início desse caso.

3.3.8 Caso de uso 8: Vectorizar dados

Este caso de uso pretende representar a vectorização dos dados de um caso.

A abordagem não paramétrica, referida anteriormente neste relatório na secção de trabalho relacionado, é deveras importante para a diferenciação dos casos.

É esta abordagem não paramétrica efectuada que vai distinguir os vários casos e permitir também a determinação da semelhança entre os mesmos.

Esta vectorização está inserida também na implementação da abordagem não paramétrica da ISA para comparação dos casos.

Como esta é uma característica muito importante para possibilitar a identificação do electrodoméstico a que pertence um caso, este requisito é classificado como **Must**.

3.3.9 Caso de uso 9: Detectar casos

Este caso de uso ilustra a situação em que são detectados casos correspondentes a um comportamento e consumo de energia efectuado por um determinado equipamento eléctrico.

Esta identificação é efectuada através da abordagem não paramétrica deste caso, referida no caso de uso anterior e do cálculo da semelhança deste caso com os vários casos guardados na plataforma.

Se existir um caso suficientemente semelhante guardado na plataforma, a semelhança entre estes dois será maior que a semelhança do caso detectado com os outros casos que também estão presentes na plataforma.

O padrão de consumo de energia do caso em questão fica desta forma associado a um equipamento eléctrico, uma vez que é determinado qual o electrodoméstico que tem este mesmo comportamento.

Estes casos são detectados numa série temporal de consumo energético obtida através do *Cloogy*.

Este é um requisito **Must**, visto que trata-se de uma das partes mais importantes da aplicação, que é a detecção de casos, através dos casos identificados e guardados anteriormente na plataforma de CBR.

3.3.10 Calcular semelhanças

Neste caso de uso é retratada a situação em que é calculada a semelhança entre o caso detectado e os casos identificados e guardados na plataforma.

Este cálculo das semelhanças entre os casos em questão é efectuado através da utilização da abordagem não paramétrica, que foi implementada pela ISA, sendo utilizada no ciclo CBR da aplicação.

O caso detectado é comparado com os casos identificados e guardados anteriormente para treinar a plataforma. Conforme a semelhança entre os casos, assim vai ser classificado este novo caso. Ou seja, o caso detectado vai ser

classificado como o comportamento de um determinado electrodoméstico, caso exista na plataforma um caso suficientemente semelhante a este.

Como este cálculo das semelhanças é uma parte importante da aplicação, resolvendo o problema de atribuição do caso detectado a um consumo energético de um determinado equipamento eléctrico, trata-se de um requisito **Must**.

3.3.11 Configurar conta

Este caso de uso pretende ilustrar a possibilidade de um utilizador configurar as suas credenciais de acesso à sua conta.

Estas credenciais permitem a comunicação entre a aplicação e a conta do *Cloogy* do respectivo utilizador.

Este é um requisito **Should**, visto que estas credenciais permitem o acesso do utilizador ao *Cloogy*, não sendo fundamental a implementação deste caso de uso para o correcto funcionamento da aplicação.

Capítulo 4

Arquitectura

4.1 Introdução

Existem vários tipos de arquitecturas de *software*. O aluno analisou alguns dos tipos de arquitecturas de *software* [13], de forma a analisar qual o mais indicado para o tipo de aplicação a desenvolver.

De seguida é apresentada e descrita a arquitectura da aplicação desenvolvida ao longo deste estágio.

4.2 Arquitectura da aplicação

Comparando os tipos de arquitectura analisados, verifica-se que a arquitectura *batch sequential* pressupõe que os seus componentes sejam executados sequencialmente, ao contrário das arquitecturas *pipe-and-filter* e *mapreduce*.

Desta forma, a arquitectura *batch sequential* tem alguma desvantagem em relação às outras, visto que não podem ser executadas tarefas paralelamente.

Neste projecto existem várias tarefas que podem ser executadas paralelamente, de modo a facilitar o processamento de dados e melhorar a performance, como é o caso dos casos de uso identificar casos, detectar casos e calcular semelhanças, podendo distribuir estas tarefas. Sendo assim, foi descartada a arquitectura *batch sequential*, em benefício das outras arquitecturas analisadas.

Comparando as arquitecturas *pipe-and-filter* e *mapreduce*, a última tem várias vantagens em relação à primeira, devido ao facto de conseguir paralelizar grandes quantidades de dados, sem grandes dificuldades para a pessoa que desenvolve o sistema, bem como o tratamento de falhas já incluído.

Devido a esta análise e ao facto de posteriormente ao estágio a empresa ISA poder utilizar a solução num programa comercial para grandes quantidades de dados, a arquitectura *mapreduce* revela-se a melhor escolha para a aplicação desenvolvida ao longo do estágio.

Desta forma, existem duas partes relevantes que correspondem às funções *map* e *reduce* da aplicação [14].

A função *map* recebe um par chave/valor, em que a chave é o nome de cada ficheiro que contém um evento e o valor é o nome da rede não paramétrica que contém os casos guardados, tendo como finalidade comparar o evento presente no ficheiro com os casos guardados na rede não paramétrica, sendo o resultado da comparação enviado para o *reduce*.

A função *reduce* agrupa todas as comparações dos eventos com os casos guardados na rede não paramétrica, que foram efectuadas na respectiva função *map*, sendo depois estes dados guardados num ficheiro pelo processo principal da aplicação.

De seguida são apresentadas algumas figuras relativas à Arquitectura da aplicação, bem como a descrição destas figuras.

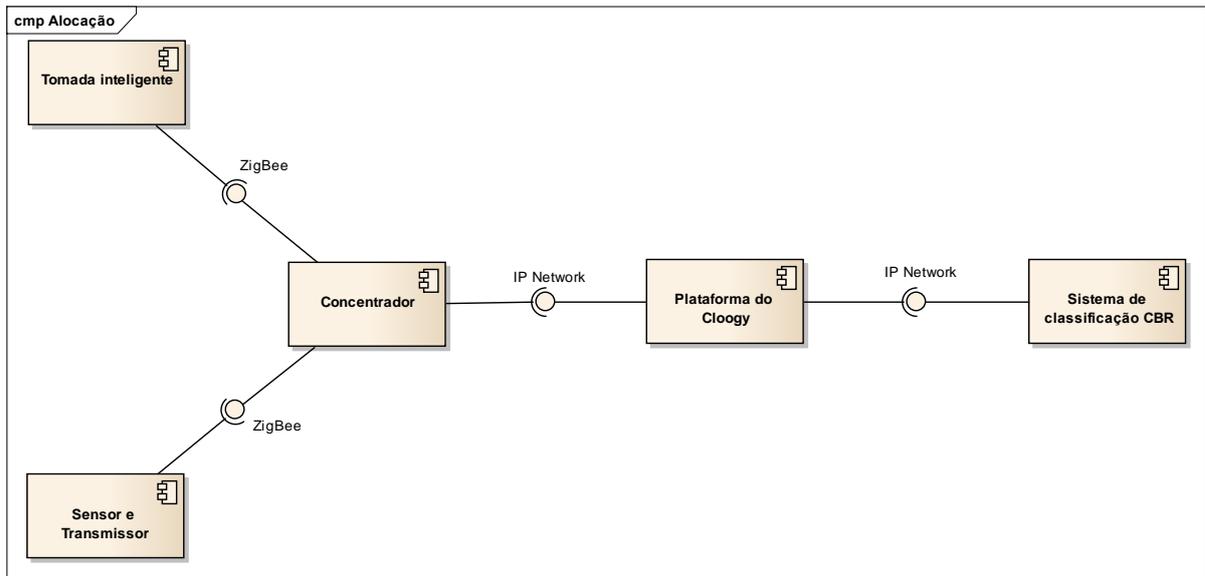


Fig. 3 – Diagrama de alocação

Na imagem anterior, está representado o diagrama de alocação.

Os componentes de alocação são elementos físicos, como é o caso das tomadas inteligentes, concentrador, sensor e transmissor. Os conectores representam os mecanismos de interacção entre os componentes físicos.

A tomada inteligente liga-se a uma tomada doméstica, sendo que os equipamentos eléctricos devem ser ligados à tomada inteligente. Esta permite que se conheça o consumo dos equipamentos domésticos ligados à mesma, bem como ligar e desligar os aparelhos à distância e agendar o seu funcionamento.

O sensor é o componente que se liga ao quadro eléctrico doméstico para recolher dados sobre os seus consumos de energia. O sensor tem de estar ligado ao transmissor, cuja função é remeter a informação recolhida pelo sensor para o concentrador.

O ZigBee representa um conjunto de especificações para a comunicação sem-fios. Este é responsável pela comunicação entre as tomadas inteligentes e o sensor e transmissor com o concentrador, através de sinais de radiofrequência não licenciados.

Por sua vez, o concentrador recolhe a informação enviada pelas tomadas inteligentes e pelo transmissor, remetendo-a para a plataforma do *Cloogy*.

O sistema de classificação CBR é a aplicação desenvolvida durante o estágio, que recebe séries temporais de consumo energético da plataforma do *Cloogy*. A comunicação entre estes dois componentes e entre o concentrador e a plataforma do *Cloogy* é efectuada através de *IP network*.

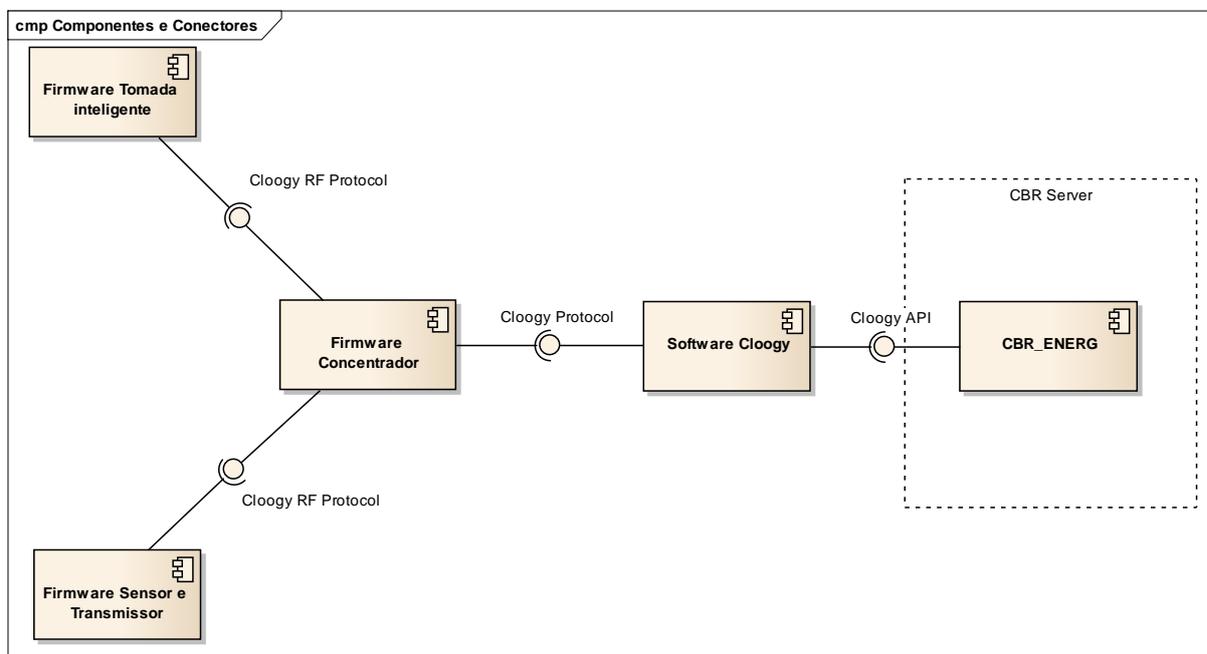


Fig. 4 – Diagrama de componentes e conectores

O diagrama de componentes e conectores está ilustrado na figura anterior.

Os *firmwares* do sensor e transmissor, tomada inteligente e do concentrador são conjuntos de instruções de operações programadas directamente sobre os respectivos dispositivos.

É utilizado um protocolo de comunicações via Rádio-Frequência, que é o *Cloogy RF Protocol*, para efectuar a comunicação entre os dispositivos descritos anteriormente, enviando para o concentrador as medições efectuadas pela tomada inteligente e pelo sensor e transmissor.

O *Software Cloogy* utiliza as funcionalidades do *Cloogy Protocol* para obter os dados das medições efectuadas, recolhidas pelo concentrador.

Por sua vez, o serviço de classificação CBR, *CBR_ENERG*, desenvolvido ao longo do estágio, utiliza as funcionalidades da *Cloogy API* para receber os dados recolhidos.

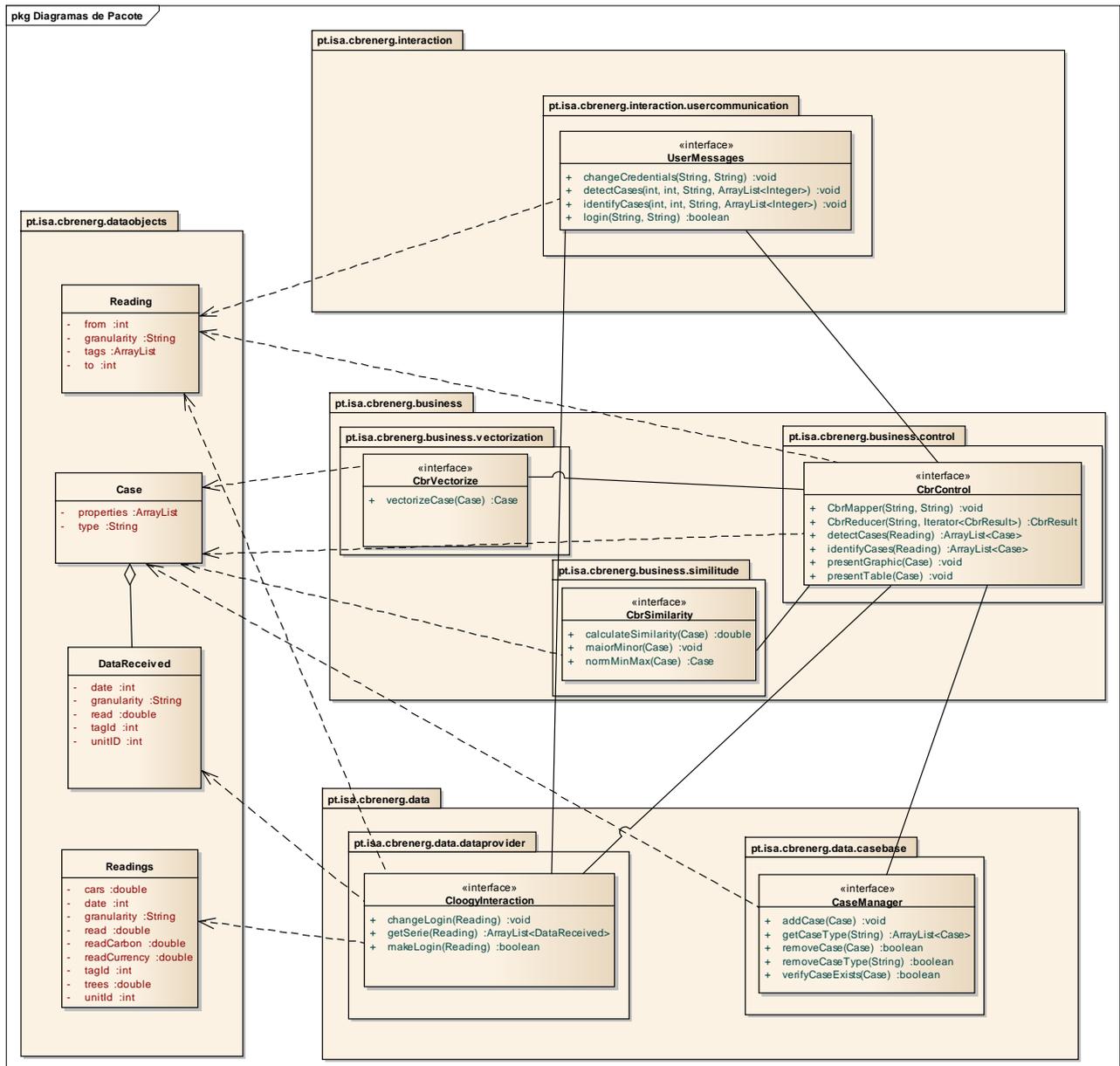


Fig. 5 – Diagrama de pacotes

Na imagem anterior está ilustrado o diagrama de pacotes da aplicação desenvolvida no estágio, sendo os mesmos descritos de seguida.

No pacote `pt.isa.cbreneng.interaction.usercommunication` está presente a interface `UserMessages` que é responsável pela comunicação com o utilizador, recebendo os pedidos efectuados pelo mesmo.

No pacote `pt.isa.cbreneng.business.control` está presente a Interface `CbrControl`, sendo que esta contém os métodos responsáveis pela gestão das funcionalidades do sistema. Os pacotes `pt.isa.cbreneng.similitude`,

pt.isa.cbrenerg.vectorization e pt.isa.cbrenerg.CaseManager fazem parte do algoritmo desenvolvido pela equipa do *Cloogy* para a abordagem não paramétrica, servindo para treino e comparação de eventos, como referido anteriormente.

O pacote pt.isa.cbrenerg.data.dataprovider tem como objectivo interagir com a *API Cloogy* para obter as séries e efectuar os pedidos do utilizador.

O pacote pt.isa.cbrenerg.dataobjects contém os objectos criados para guardar informação dos casos, dados obtidos da *API Cloogy* e pedidos efectuados a esta mesma API.

A aplicação desenvolvida é independente dos algoritmos que se pretendam utilizar para o treino e comparação de eventos, sendo possível e simples usar outros algoritmos utilizados para estas finalidades.

Capítulo 5

Análise de dados efectuada

5.1 Recolha de dados

A recolha de dados efectuada pela aplicação é realizada com recurso ao *Cloogy*, que é a solução actual da ISA para monitorização da energia em ambientes domésticos, através de um *clamp* colocado à entrada do quadro eléctrico e da monitorização e actuação em tomadas individuais usando tomadas inteligentes.

A taxa de amostragem dos dados recolhidos é de minuto em minuto.

O *Cloogy* é uma solução de gestão energética que permite aos clientes da ISA monitorizar e controlar o consumo total de casa ou escritório.

Consiste num sistema integrado que combina dispositivos de recolha de dados com plataformas de visualização e controlo, disponíveis através do computador do cliente, *smartphone*, *tablet* ou de um monitor produzido pela ISA para este efeito.

O *Cloogy* abre a porta da sustentabilidade, permitindo ao cliente desta solução saber onde a casa ou escritório são ineficientes e onde poupar energia. Desta forma, é possível otimizar a utilização dos equipamentos eléctricos, reduzir desperdícios e custos desnecessários, contribuindo assim para uma vida mais sustentável.

É ainda possível ao cliente monitorizar também o consumo individual dos seus equipamentos eléctricos, permitindo controlar o horário de funcionamento de cada um, através das tomadas inteligentes.

A ISA oferece aos seus clientes várias formas possíveis de utilizar o *Cloogy*, consoante seja para casa ou escritório a solução pretendida, sendo que as diferenças entre as várias modalidades são verificadas na quantidade de equipamentos disponibilizados, que se reflectem consequentemente no preço da solução.

Esta solução de monitorização de energia consiste num sensor instalável no contador de electricidade, que vai recolher dados sobre os consumos gerais da casa ou escritório do utilizador. As tomadas inteligentes enviam informação sobre o consumo dos equipamentos ligados às mesmas. Uma vez que as tomadas inteligentes podem ser mudadas de sítio, passando a reportar o consumo energético de outro equipamento, é necessário a identificação do aparelho eléctrico da tomada. Todos os dados são enviados, posteriormente, para um concentrador, que os envia para as várias plataformas de monitorização.

Através destas plataformas é possível ligar ou desligar os equipamentos eléctricos à distância e agendar o funcionamento dos mesmos.

Para o funcionamento do *Cloogy* é necessário um ponto fixo de acesso à Internet (ADSL, Cabo, Fibra, Wi-Fi), sendo que o serviço não suporta pontos de acesso móvel, como é o caso de placas portáteis de banda larga 3G.

A imagem seguinte ilustra o funcionamento do sistema descrito anteriormente.

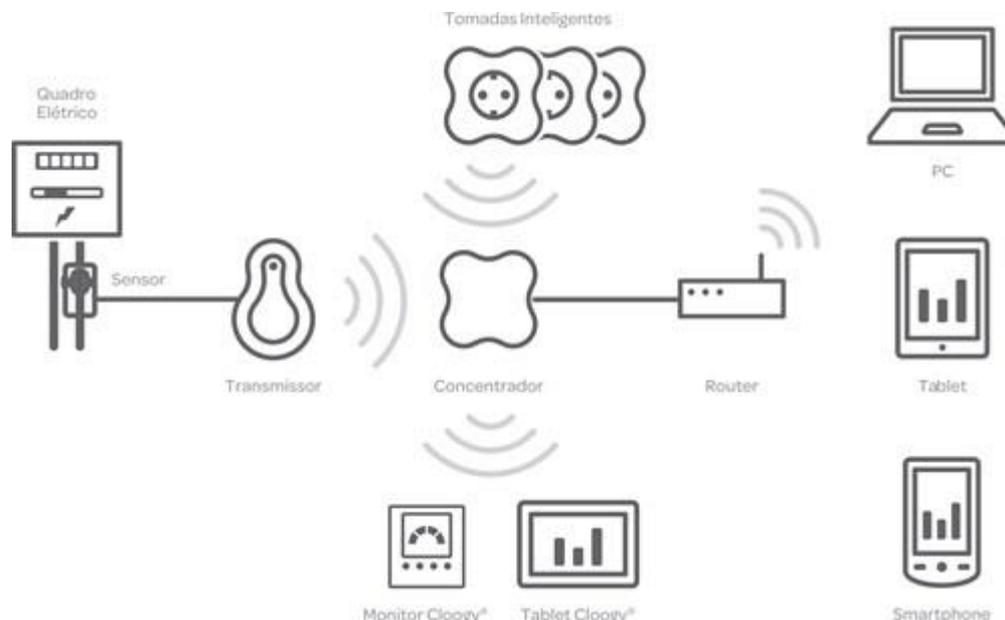


Fig. 6 – Funcionamento do *Cloogy*

5.2 Abordagens utilizadas

5.2.1 Identificação de início de eventos

As séries temporais de consumo energético recebidas pela aplicação, através da API *Cloogy*, fornecem o consumo de energia para cada minuto.

O primeiro problema que se pretendia resolver era a identificação automática do início de eventos, uma vez que na maior parte do tempo não existem alterações significativas nos valores medidos.

A identificação de possíveis inícios de eventos foi feita aplicando dois filtros sucessivos às séries temporais de consumo energético, onde é pretendido identificar início de eventos, seleccionando posteriormente apenas os valores situados acima de um determinado limite.

Desta forma é aplicada uma média móvel com a fórmula:

$$Y(n) = \frac{X(n-1)+X(n)+X(n+1)}{3} \quad (5.1)$$

O segundo filtro aplicado tem a seguinte fórmula:

$$W(n) = \frac{Y(n+1)-Y(n)}{2} \quad (5.2)$$

Depois de aplicar estes dois filtros, a selecção dos valores é efectuada segundo a regra:

$$Z(n) = \begin{cases} 1 & \text{se } W(n) \geq \text{limite} \\ 0 & \text{se } W(n) < \text{limite} \end{cases} \quad (5.3)$$

Potência
(KW)

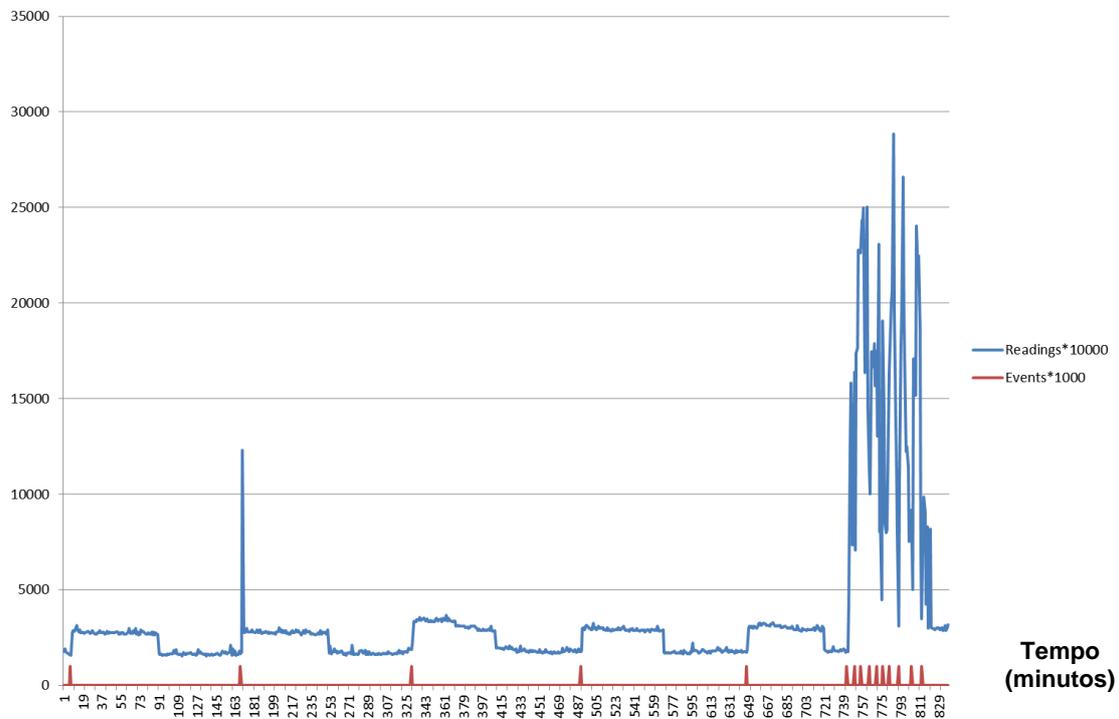


Fig. 7 – Identificação de início de eventos numa série temporal de consumo energético

O resultado do processo de identificação de início de eventos está ilustrado na figura 7, onde o processo foi aplicado a uma série temporal de consumo energético. Numa questão de representação, os valores da série temporal de consumo energético foram multiplicados por 10000 e o início de eventos por 1000 de modo a que as duas linhas não se sobreponham.

O valor utilizado para o limite foi 100, tendo sido identificado numa abordagem tentativa-erro, permitindo a detecção de início de eventos conhecidos mas não é sensível ao ponto de não detectar pequenas alterações de consumo energético.

Após os 721 minutos da série temporal de consumo energético ilustrada no gráfico, verifica-se que são identificados vários possíveis inícios de eventos, sendo este o comportamento esperado porque o objectivo é identificar possíveis inícios de eventos para accionar a identificação de eventos através da plataforma CBR.

A validação do processo foi feita recorrendo à inspecção visual de séries temporais de consumo energético relativas a 20 dias.

5.2.2 Abordagem estatística

A abordagem estatística utilizada na aplicação realizada neste estágio foi efectuada utilizando um vector com cinco características fáceis e rápidas de calcular, validar e que podiam ser utilizadas para testar o funcionamento da plataforma de case-based reasoning.

As características utilizadas são:

- **Sum** – somatório da energia consumida no período de tempo do evento;
- **Avg** – média do consumo de energia realizado no evento;
- **Std** – desvio padrão do consumo de energia no período de tempo do evento;
- **Min** – valor mínimo de energia consumida no evento;
- **Max** – valor máximo de energia consumida no evento.

Embora as características descritas anteriormente sejam simples medidas estatísticas, estas foram utilizadas para comprovar que o ciclo CBR estava a funcionar como era esperado.

5.2.3 Abordagem não paramétrica

Visto que a abordagem estatística efectuada revelou ser insuficiente para a correcta identificação dos eventos ocorridos, foi testada a hipótese de utilizar uma descrição não paramétrica da série temporal de consumo energético de modo a detectar melhor as alterações ocorridas nesta.

Esta abordagem foi desenvolvida pela equipa do *Cloogy*, sendo que o mestrando utiliza-a para a classificação de eventos, bem como para o treino do ciclo CBR com os casos guardados.

Como ponto de partida decidiu-se utilizar a mesma resolução dos dados extraídos da API do *Cloogy*, ou seja com uma taxa de amostragem de minuto

em minuto e, uma vez provado que esta resolução permite a identificação dos consumos desejados, reduzir o número de pontos no vector.

Para cada ponto foi calculada a média e o desvio padrão de todos os valores na mesma hora e nos minutos adjacentes num intervalo chamado janela. Numa janela com tamanho 0 só é utilizado um minuto e numa janela com tamanho 1 são utilizados os minutos $n-1$, n e $n+1$.

O treino foi efectuado utilizando as leituras da tomada inteligente onde a máquina de lavar a louça está ligada e do concentrador que está ligado ao quadro eléctrico da casa. As figuras seguintes fornecem uma representação gráfica da média calculada para cada caso, sendo que a média é representada pela curva MED e cada caso é representado por uma curva Ex.

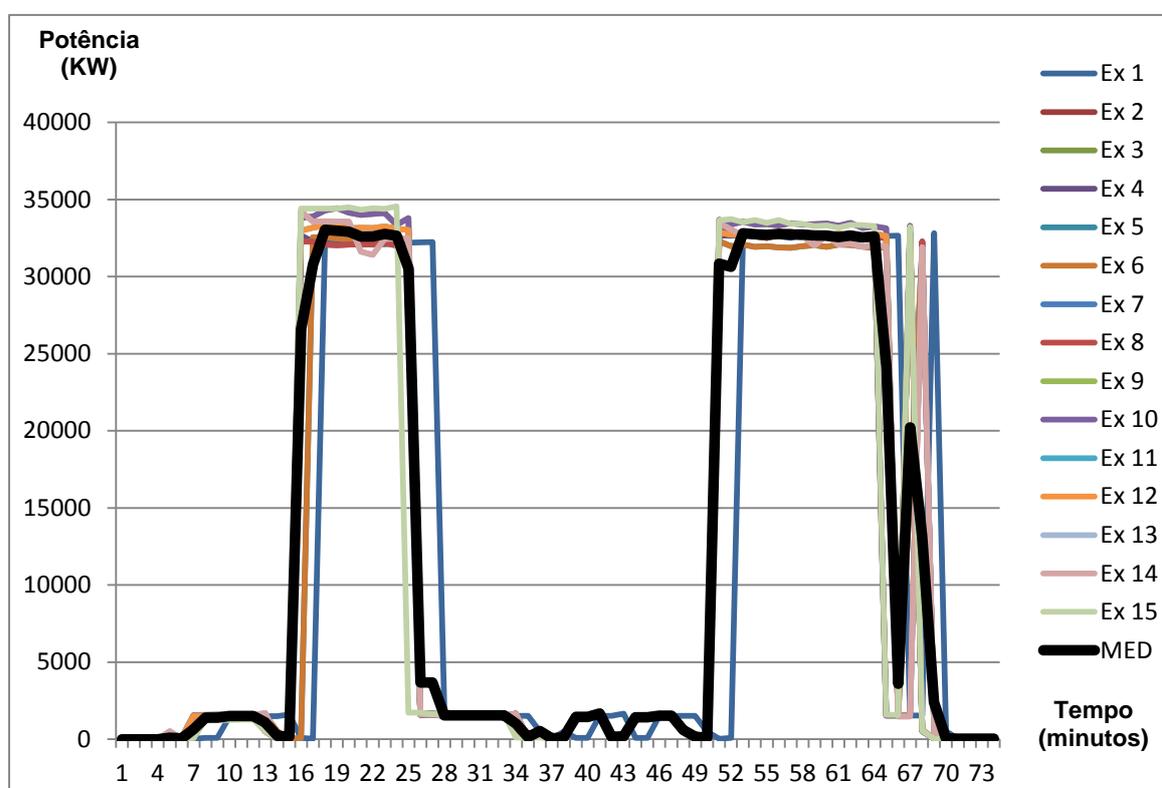


Fig. 8 – 15 leituras da tomada inteligente e a média destas, considerando uma janela de tamanho 0

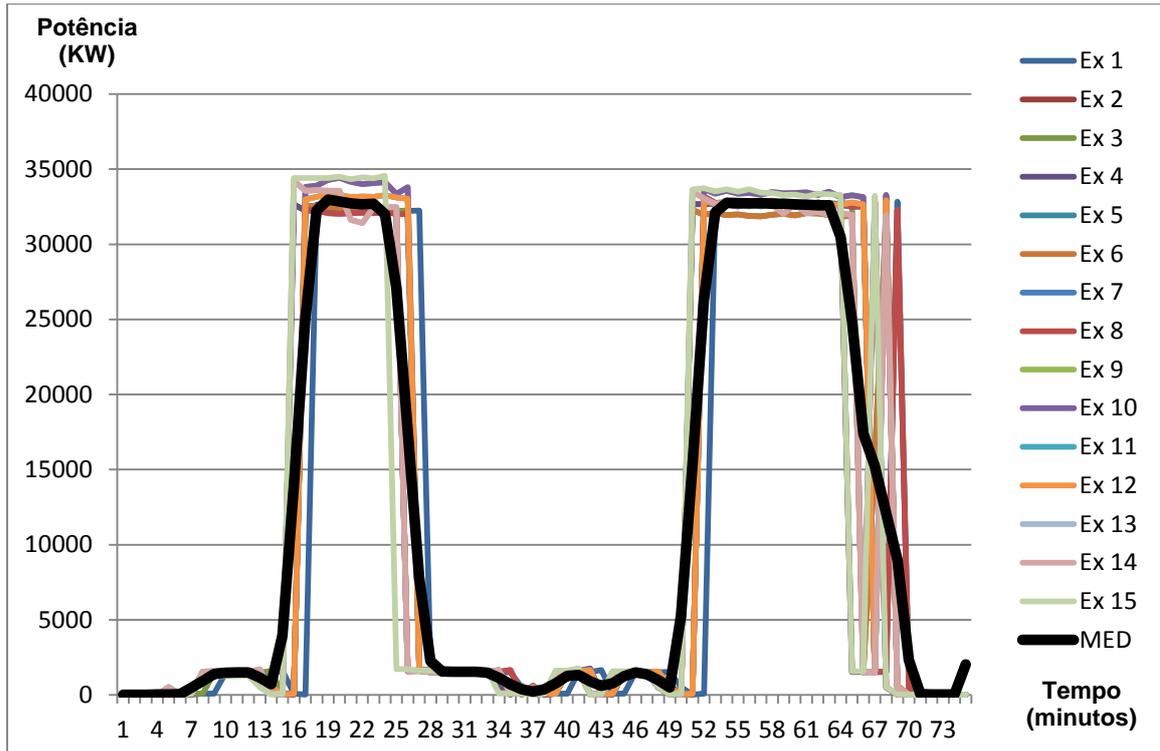


Fig. 9 – 15 leituras da tomada inteligente e a média destas, considerando uma janela de tamanho 1

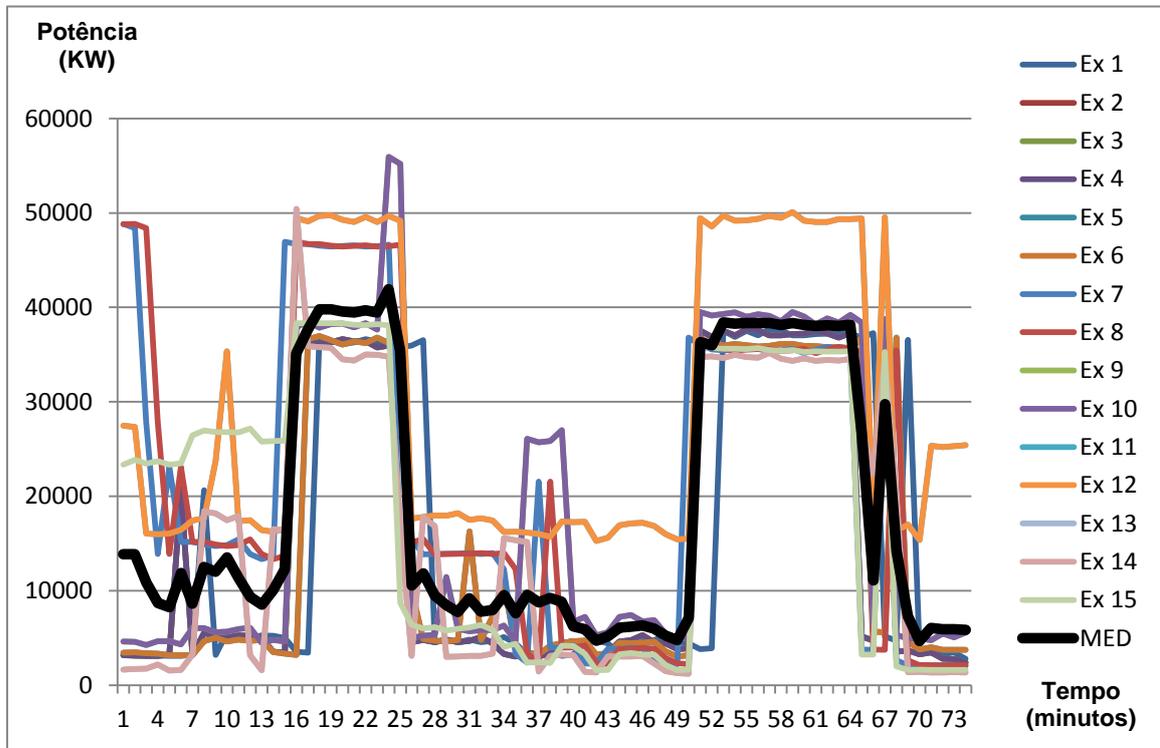


Fig. 10 – 15 leituras do concentrador e a média destas, considerando uma janela de tamanho 0

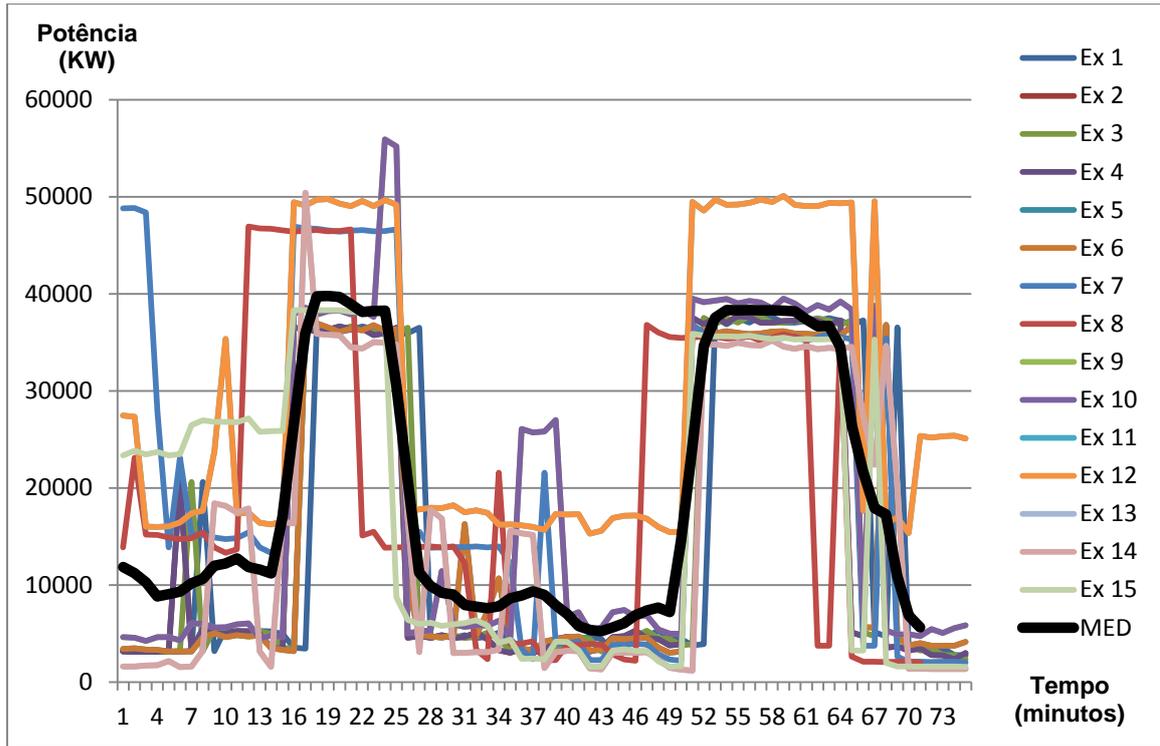


Fig. 11 – 15 leituras do concentrador e a média destas, considerando uma janela de tamanho 1

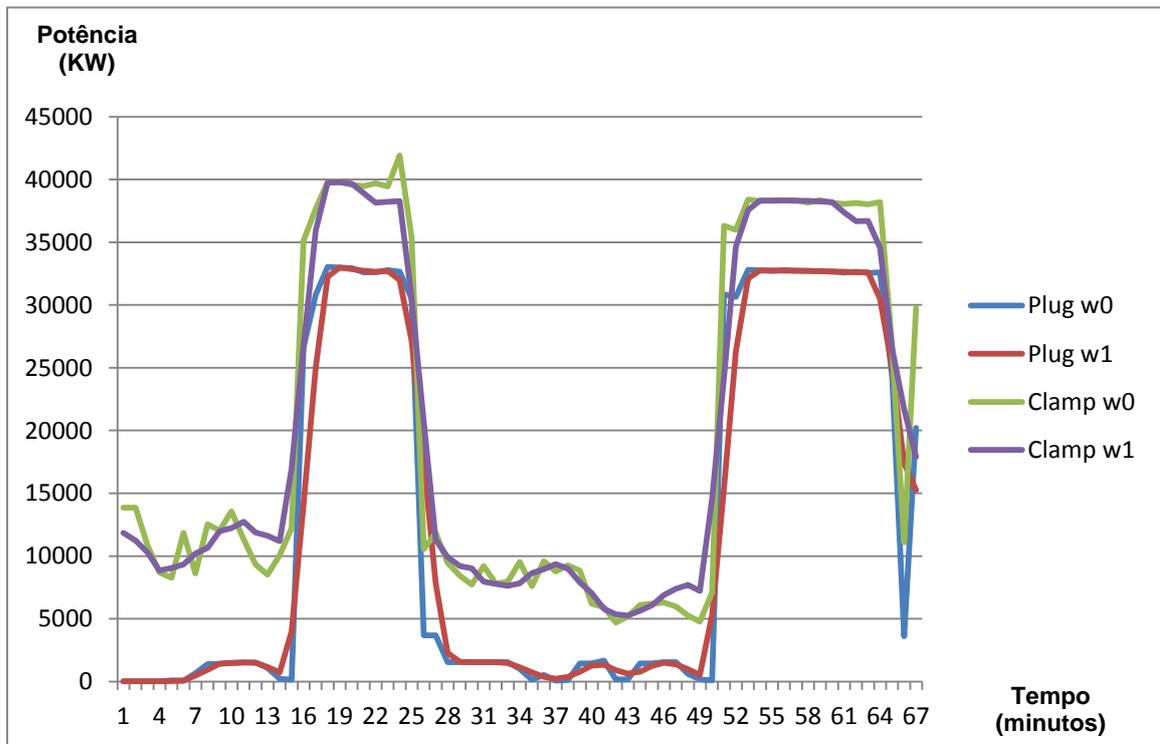


Fig. 12 – Comparação de todas médias calculadas

Como esperado, as curvas com janela de tamanho 1 têm menos resolução do que as curvas com janela de tamanho 0, o que pode ter impacto na identificação dos padrões. Outra diferença importante que se verifica é o deslocamento superior dos gráficos do concentrador em relação aos gráficos da tomada inteligente, visto que existem mais consumos de energia ao mesmo tempo.

Todos os pontos da série temporal de consumo energético são normalizados num valor entre 0 e 1. Posteriormente, a função de semelhança utilizada é calculada determinando a média da semelhança de cada ponto.

Seguindo as seguintes considerações:

- **avg** – média no ponto x ;
- **sdv** – desvio padrão no ponto x ;
- **value** – valor a testar;
- **density(x)** – retorna a função densidade de probabilidade de uma distribuição com média 0 e desvio padrão 1, no ponto x .

Obtém-se a função de semelhança entre o valor *value* e a média *avg*:

$$Sim(avg, sdv, value) = density\left(\frac{|value-avg|}{sdv}\right) \quad (5.4)$$

Capítulo 6 Planeamento

Existem várias metodologias que podem ser utilizadas para o planeamento de um projecto de *software*.

O trabalho a efectuar durante o segundo semestre deste estágio é efectuado segundo o Processo de Concepção e Desenvolvimento da ISA, isto porque as directrizes da ISA obrigam o aluno a seguir este processo, visto que a empresa tem normas a seguir. As normas que a empresa segue e está certificada com estas são a ISO 9001 de gestão de qualidade, a NP 4457 de gestão de investigação, desenvolvimento e inovação e a CMMI-DEV de desenvolvimento de *software* de nível 2 de maturidade.

Este processo de concepção e desenvolvimento contém algumas ideias da metodologia **SCRUM** [15].

De seguida é apresentada uma imagem que ilustra as várias fases do processo de concepção e desenvolvimento da ISA, estando presente nos anexos confidenciais documentos que descrevem este mesmo processo.

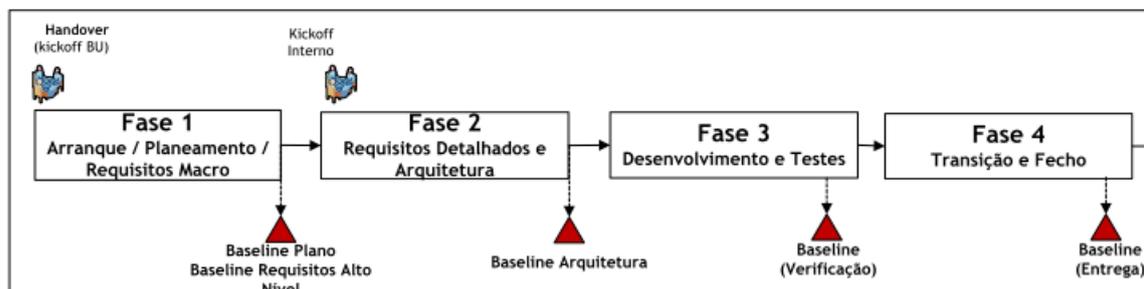


Fig. 13 – Processo de Concepção e Desenvolvimento da ISA

A metodologia SCRUM consiste em estruturar o trabalho em iterações, chamadas *sprints*.

Estas iterações são normalmente menores que um mês, ocorrendo umas após as outras, começando e acabando numa data específica, quer o trabalho esteja concluído ou não, nunca sendo estendidas.

No início de cada *sprint*, existe uma reunião para se decidir o trabalho a realizar até ao final do mesmo, não mudando o trabalho ao longo da iteração. Sendo que no final da iteração é analisado o trabalho efectuado.

Desta forma, foram estabelecidos *sprints* que têm a duração de 3 semanas cada um, em que no início de cada *sprint* há uma reunião com o orientador da

empresa, na qual são estabelecidos os objectivos e o que deve ser realizado no mesmo *sprint*.

Consequentemente, existem assim 7 *sprints* de 3 semanas cada um, dividindo o tempo de trabalho do segundo semestre.

O artigo, que é referido na figura 4 deste relatório, que contempla a divisão do trabalho no segundo semestre, é um artigo para ser apresentado na conferência “International Workshop Case-Based Reasoning CBR-MD 2013”, em New York, USA sobre CBR da Industrial Conference on Data Mining, sendo que o mesmo já foi submetido. Este artigo está a ser planeado pela ISA há dois anos, sendo que o aluno colaborou no mesmo, com as experiências efectuadas até à data de submissão do artigo.

6.1 Planeamento inicial

Em cada *sprint* foi efectuado planeamento, desenvolvimento, validação e verificação para o trabalho a realizar nesse mesmo *sprint*.

No primeiro *sprint* estava planeado continuar as experiências já referidas no capítulo anterior para a publicação do artigo, aplicando a experiência efectuada a uma casa inteira que tem um sensor no quadro e sete tomadas inteligentes. Pretendendo assim verificar se os eventos detectados nas tomadas, ou seja, nos sinais isolados dos electrodomésticos também são detectados no sinal do quadro, em que está presente o sinal geral do consumo de energia da habitação.

O segundo *sprint* consiste na colaboração da redacção do artigo e do fecho da arquitectura.

No terceiro e quarto *sprint* pretendia-se que a aplicação fosse desenvolvida, sendo esta solução testada e validada com dados de mais casas no seguinte *sprint*, ou seja o quinto.

De 22 de Maio a 14 de Junho, isto é, no sexto e penúltimo *sprint* estava planeado efectuar correcções e escrever o relatório final de estágio, enquanto que o último *sprint*, compreendido entre 17 de Junho e 08 de Julho seria dedicado à elaboração da apresentação do estágio.

De seguida é apresentada uma imagem que ilustra a divisão do tempo de trabalho em *sprints*, referida anteriormente.

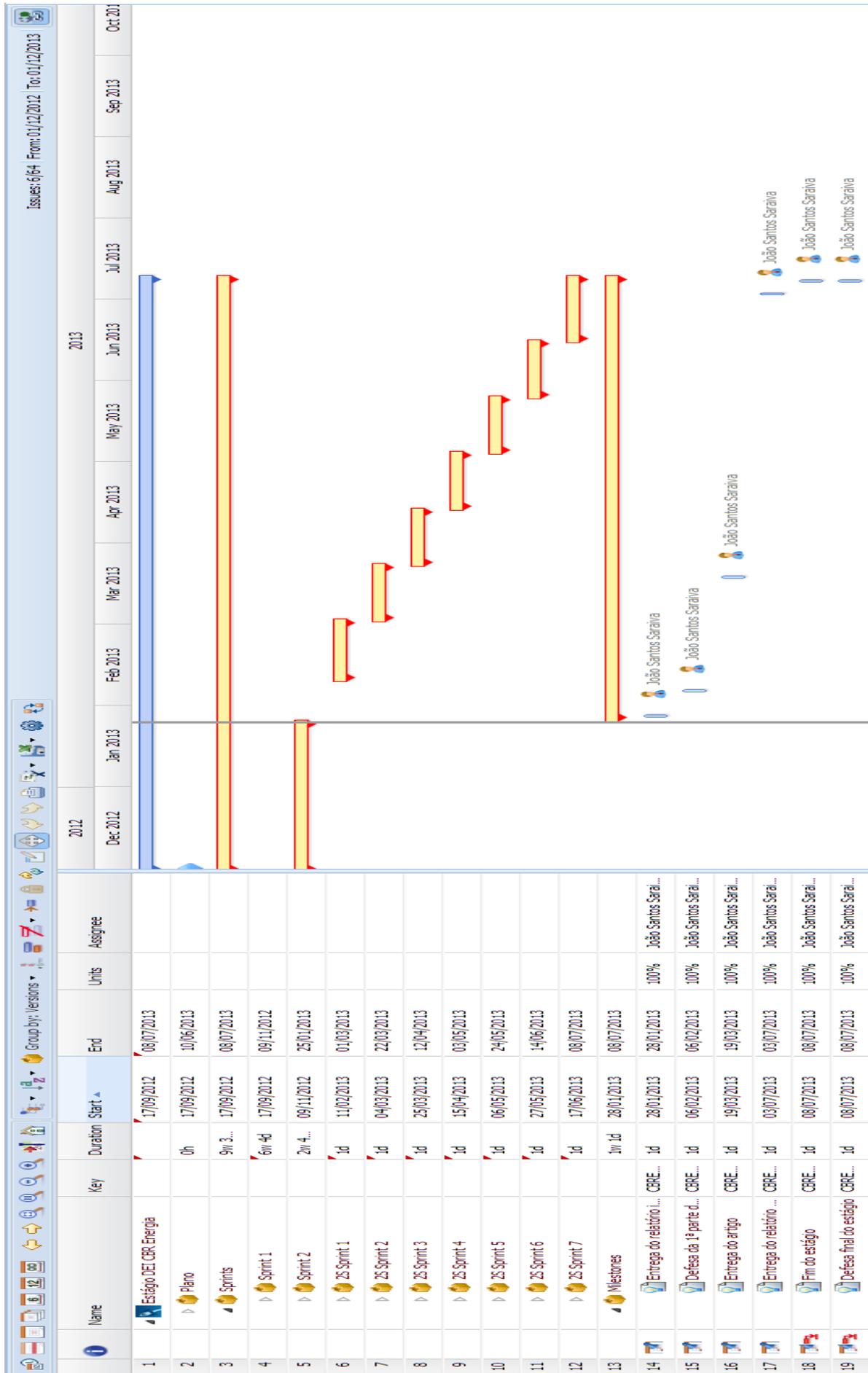


Fig. 14 – Planeamento do segundo semestre

6.2 Reflexões das iterações

6.2.1 Iteração 1 (11/02/2013 a 01/03/2013)

Nesta iteração foram realizadas alterações ao relatório, segundo as observações e indicações do júri na defesa intermédia do estágio.

Foram também continuadas as experiências efectuadas anteriormente no primeiro semestre para detecção do início de eventos.

A realização das tarefas efectuadas nesta iteração demorou 118 horas.

6.2.2 Iteração 2 (04/03/2013 a 22/03/2013)

As tarefas realizadas nesta iteração prendem-se com a elaboração da Arquitectura inicial da aplicação, bem como alteração de requisitos e experiências de detecção de início de eventos e comparação de eventos para inclusão no artigo científico.

Esta iteração teve 114 horas de trabalho para a realização das tarefas referidas.

6.2.3 Iteração 3 (25/03/2013 a 12/04/2013)

A iteração 3 foi realizada em 116 horas de trabalho.

As tarefas realizadas foram a leitura de um capítulo do livro [14], referente a estilos de Arquitecturas (Batch Sequential, Pipe-and-filter e MapReduce), escrita do capítulo de Arquitectura no relatório de estágio, fecho da Arquitectura e início da elaboração do desenho detalhado.

6.2.4 Iteração 4 (15/04/2013 a 03/05/2013)

Na iteração 4 foi concluído o desenho detalhado, leitura de artigos científicos, alteração do relatório de estágio e início de implementação dos interfaces java para elaboração da aplicação.

Foi ainda realizada uma reunião entre o aluno, o orientador do DEI e o orientador da ISA, como é sugerido na plataforma de estágios.

O mestrando trabalhou 115 horas nesta iteração para realizar as tarefas referidas.

6.2.5 Iteração 5 (06/05/2013 a 24/05/2013)

Nesta iteração foram implementadas as interfaces java, iniciou-se a implementação do ciclo CBR da aplicação e foram efectuados tutoriais de jboss e infinispn para a implementação da arquitectura MapReduce.

A realização das tarefas efectuadas nesta iteração demorou 114 horas.

6.2.6 Iteração 6 (27/05/2013 a 14/06/2013)

As tarefas realizadas nesta iteração prendem-se com o fim da implementação do ciclo CBR da aplicação, bem como a integração na arquitectura MapReduce elaborada.

Esta iteração teve 117 horas de trabalho para a realização das tarefas referidas.

6.2.7 Iteração 7 (17/06/2013 a 08/07/2013)

As tarefas realizadas foram a escrita do relatório final, elaboração da apresentação final de estágio e preparação da apresentação.

Até à data de entrega deste relatório, as tarefas realizadas nesta última iteração perfazem um total de 108 horas, estando ainda planeadas mais 12 horas até ao fim da mesma iteração, fazendo um total de 120 horas de trabalho para a realização destas tarefas.

6.2.8 Conclusão

As iterações descritas nas secções anteriores são referentes ao trabalho realizado no segundo semestre, perfazendo um total de 814 horas de esforço para o mestrando, sendo que o esforço previsto para este semestre da cadeira de Dissertação/Estágio são 810 horas.

Capítulo 7

Desenvolvimento

7.1 Implementação

Relativamente à implementação da aplicação desenvolvida durante o presente estágio, importa focar nas duas partes essenciais desta, que são o ciclo CBR e o MapReduce, sendo a sua implementação descrita nas subsecções que se seguem.

7.1.1 Ciclo CBR

O ciclo CBR é constituído pelas suas quatro fases, como é referido na secção 2.1 deste relatório de estágio. Desta forma, importa descrever cada fase do ciclo CBR da aplicação implementada.

Na primeira fase, que tem o nome de *Retrieve*, a aplicação obtém todos os casos guardados no treino efectuado, colocando a rede não paramétrica que os contém em *cache*, de forma a poder comparar todos os eventos pretendidos com os casos guardados.

Seguidamente, na fase *Reuse*, os eventos que se pretendem classificar são comparados com os casos guardados na rede não paramétrica, sendo que para cada evento existe um *map*, permitindo executar a comparação de todos os eventos paralelamente, como é explicado na subsecção seguinte.

O objectivo é que a fase *Revise* seja efectuada pelo utilizador no produto final implementado pela ISA, ao dar *feedback* para a aplicação de quando realmente utilizou um equipamento, de modo a confirmar as identificações de eventos efectuadas pela abordagem não paramétrica que é utilizada na fase *Reuse*.

Posteriormente são seleccionados alguns dos eventos para adicionar aos casos guardados, sendo estes os falsos negativos, visto que a aplicação não conseguiu identificar estes eventos como funcionamento do equipamento eléctrico em específico. Desta forma, quando surgir um consumo semelhante, a aplicação já não vai ter este falso negativo, passando a identificar correctamente o evento.

7.1.2 MapReduce

O MapReduce implementado tem como objectivo realizar a comparação de vários eventos paralelamente, dividindo-se em duas operações: o *map* e o *reduce*. De seguida é explicado como foi realizada a sua implementação.

A função *map* recebe um par chave/valor, em que a chave é o nome de cada ficheiro que contém um evento e o valor é o nome da rede não paramétrica que contém os casos guardados, tendo como finalidade comparar o evento presente no ficheiro com os casos guardados na rede não paramétrica, sendo o resultado da comparação enviado para o *reduce*.

A função *reduce* agrupa todas as comparações dos eventos com os casos guardados na rede não paramétrica, que foram efectuadas na respectiva função *map*, sendo depois estes dados guardados num ficheiro pelo processo principal da aplicação.

7.2 Dificuldades encontradas

Durante a implementação da aplicação desenvolvida ao longo do estágio, o mestrando deparou-se com alguns desafios.

Os desafios iniciais deveram-se ao facto de desconhecer algumas ferramentas com as quais necessitava de trabalhar, como é o caso do jBoss AS, o Infinispan e o Code Pro AnalytiX da Google para testes do código desenvolvido.

O desconhecimento das referidas ferramentas levou a uma curva de aprendizagem através de tutoriais sobre as mesmas, o que atrasou o início da implementação da aplicação.

Durante a implementação da aplicação surgiram também algumas dificuldades relacionadas com o Infinispan, que é uma plataforma presente no jBoss que possibilita a criação de um ambiente *cluster* de uma forma simples, bem como o acesso à cache de forma transparente para o utilizador. Permite também a implementação do MapReduce sobre o servidor jBoss, não tendo o utilizador de se preocupar com a forma de como é distribuída a execução do mesmo.

Estas dificuldades surgiram devido à falta de documentação em relação ao MapReduce, sendo que os tutoriais existentes são muito semelhantes e a informação relativa às funções da biblioteca do Infinispan, presentes no *site* do jBoss, não eram muito elucidativas. Uma vez que as funções *map* e *reduce* dos tutoriais apenas recebiam *strings* e *integers* e a própria implementação da função na biblioteca do Infinispan obrigava a este tipo de parâmetros, demorou algum tempo para o mestrando conseguir implementar as funções referidas com objectos *java* criados pelo aluno.

A utilização da cache também não foi trivial, uma vez que a rede não paramétrica tinha de ser acessível por todos os *mappers*, para cada um conseguir fazer a comparação do seu evento com os eventos presentes na rede não paramétrica.

Capítulo 8

Testes

Durante a implementação da aplicação foram também realizados vários testes à mesma.

Foram realizados testes unitários a todas as funções da aplicação, para garantir que todas as funções estavam a funcionar de acordo com o pretendido, verificando que ao receber os valores determinados no teste, produziam os resultados esperados para esses valores.

Para realizar estes testes foi utilizada a ferramenta Code Pro AnalytiX da Google, sendo esta a ferramenta utilizada em todos os projectos de *software* da ISA, seguindo assim o processo de Concepção e Desenvolvimento da empresa.

Foram ainda efectuados outros testes, como é o caso de executar a aplicação nas duas formas de *deployment* do jBoss 7.1, que são a *standalone* e a *domain*. Este teste foi de extrema importância, visto que na forma *standalone*, o projecto é executado localmente, enquanto que na forma *domain* o mesmo é executado num grupo de servidores. Ao conseguir executar o projecto no grupo de servidores, conseguiu-se provar que a aplicação desenvolvida pode ser executada futuramente num *cluster*, como a empresa pretende. O teste também contribuiu para a detecção de um erro no código da abordagem não paramétrica, efectuada pela ISA, sendo que algumas classes java implementadas, utilizadas no MapReduce não eram *Serializable*, não podendo dessa forma ser utilizadas paralelamente pelos vários *mappers*.

Os requisitos identificados inicialmente foram cumpridos, uma vez que a ISA validou e aceitou a aplicação desenvolvida ao longo do estágio.

Capítulo 9

Conclusões e Trabalho futuro

9.1 Conclusões

Criar uma aplicação de *software* para um cliente é um grande desafio para um Engenheiro de *Software*.

Este estágio foi uma oportunidade de experimentar todas as fases do desenvolvimento de um produto de *software*, desde pensar na solução na fase de levantamento de requisitos, passando pela integração com tecnologias já existentes e terminando na fase de teste da aplicação.

Achei interessante e gratificante o facto de trabalhar com ferramentas e tecnologias com as quais ainda não tinha tido contacto, como é o caso do Code Pro AnalytiX da Google, o jBoss AS e o Infinispan. A curva de aprendizagem destas ferramentas e tecnologias tinha de ser obrigatoriamente curta, para a possível implementação do sistema, existindo de certeza muitas mais funcionalidades por aprender nas mesmas.

Concluindo, este estágio fez-me melhorar em termos de Engenharia de *Software*, oferecendo-me uma experiência de todas as fases de desenvolvimento de *software*.

9.2 Trabalho futuro

Existem várias hipóteses que podem ser consideradas para o futuro dando seguimento ao trabalho realizado neste estágio.

A aplicação produzida pode ser integrada no produto *Cloogy* da ISA e dar ao utilizador dicas sobre quando deveria um equipamento trabalhar para poder poupar energia e conseqüentemente os seus gastos financeiros com a factura da electricidade. Por exemplo, se um utilizador tem uma tarifa bi-horária, o *Cloogy* em conjunto com a aplicação ao detectar a utilização da máquina de lavar a roupa num horário em que o preço de kWh é mais caro podia aconselhar o cliente a mudar a utilização do electrodoméstico para um horário mais adequado.

Outras hipóteses são também a implementação num *cluster*, com distribuição de ficheiros, bem como a inclusão da identificação de novos equipamentos eléctricos na solução encontrada.

Bibliografia

- [1] AAMODT, A., AND PLAZA, E. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI communications* 7, 39-59.
- [2] PAL, S.K., AND SHIU, S.C.K. 2004. Foundations of Soft Case-Based Reasoning. Wiley Series on Intelligent Systems, New Jersey, USA.
- [3] WATSON, I., AND MARIR, F. 1994. Case-Based Reasoning: A Review. *Knowledge Engineering Review* 9, 327-354.
- [4] RIESBECK, C., AND SCHANK, R. 1989. Inside Case-Based Reasoning. *Erlbaum*.
- [5] WATSON, I. 1999. Case-Based Reasoning is a Methodology not a Technology. *Knowledge-Based Systems* 12, 303-308.
- [6] OLSSON, E., AND FUNK, P., AND XIONG, N. 2004. Fault Diagnosis in Industry Using Sensor Readings and Case-Based Reasoning. *Journal of Intelligent and Fuzzy Systems* 15, 41-46.
- [7] CLARKSON, K.L. 2006. Nearest-Neighbor Searching and Metric Space Dimensions. *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice* 15-59.
- [8] COSTA, E., AND SIMÕES, A. 2008. Inteligência Artificial Fundamentos e Aplicações. FCA - Editora de Informática, Coimbra, Portugal.
- [9] WEISS, M., AND HELFENSTEIN, A., AND FRIEDEMANN, M., AND STAAKE, T. 2012. Leveraging Smart Meter Data to Recognize Home Appliances. In *Pervasive Computing and Communications (PerCom) on IEEE International Conference*, Lugano, Switzerland, March 2012, IEEE, 190-197.
- [10] HUGHES, T., AND LINGE, N., AND HILL, S. 2012. Personalised Energy Monitoring Systems.
- [11] MALLYA, S. 2011. Entracker: Energy Tracker for Homes.
- [12] BRENNAN, K. 2009. A Guide to the Business Analysis Body of Knowledge (BABOK guide), version 2.0. liba, Toronto, Canada.
- [13] ZHU, H. 2005. Software Design Methodology: From Principles to Architectural Styles. Butterworth-Heinemann, Oxford, UK.
- [14] DEAN, J., AND GHEMAWAT, S. 2008. MapReduce: Simplified Data

Processing on Large Clusters. *Communications of the ACM* 51, 107-113.

[15] SUTHERLAND, J. 2010. Scrum Handbook. Scrum Training Institute Press, Boston, USA.