João Pedro Reis Marques Henriques Rainho

Projeto SOCIAL - Desenvolvimento de componente para a deteção de eventos significativos em tempo real

Relatório Final de Estágio
Mestrado em Engenharia Informática
orientada por Porf. Alberto Cardoso e António Damasceno
e apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Julho de 2018

· U C ·

Universidade de Coimbra

João Pedro Reis Marques Henriques Rainho

Projeto SOCIAL - Desenvolvimento de componente para a deteção de eventos significativos em tempo real

Relatório Final de Estágio
Mestrado em Engenharia Informática
orientada por Prof. Alberto Cardoso e António Damasceno
e apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra
Júri: Prof. Edmundo Monteiro e Prof. Paulo Rupino

Julho de 2018

Universidade de Coimbra

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus orientadores. Ao António Damasceno, pelo apoio diário, transmitindo muita da sua e experiência e dando-me a oportunidade de integrar no projeto SOCIAL e ao Prof. Alberto Cardoso pelo conhecimento e suporte dado ao longo deste último ano.

A todo o consórcio do projeto, que ajudou a esclarecer algumas questões ao longo deste estágio, mas particularmente ao André e ao Telmo pelo apoio diário dado no IPN.

Gostaria de agradecer também aos meus colegas de curso, que proporcionaram grandes momentos de alegria e diversão e ajudaram-me a ultrapassar várias etapas ao longo da minha vida académica.

Aos meus amigos, que continuaram a mostrar que posso sempre contar com eles e à Sara que foi um dos meus grandes suportes nos momentos mais difíceis.

Por último, mas não menos importante, tenho que agradecer à minha família, em particular aos meus pais, à minha irmã e aos meus padrinhos, que sempre acreditaram em mim e por me proporcionarem a formação e educação necessária para que o trabalho que aqui apresento fosse possível.

Resumo

O envelhecimento da população é um dos grandes desafios que o mundo vive atualmente. O aumento da esperança média de vida é uma das principais causas das alterações demográficas e socioeconómicas e, consequentemente, do aumento do número de idosos. Por isso, é necessário dar uma maior atenção a esta faixa etária, de forma a diminuir o risco de exclusão social.

Devido à falta de soluções tecnológicas que garantam uma interoperabilidade entre as várias instituições de ação social, é necessário desenvolver os sistemas de informação de forma a melhorar a comunicação entre os vários intervenientes. A generalização das soluções de IoT e o aumento das capacidades integradoras dos sistemas podem garantir uma maior proximidade de monitorização e avaliação do cidadão.

Para colmatar estes problemas surge a plataforma SOCIAL (*Social Cooperation for Integrated Assisted Living*), que pretende melhorar a integração e continuidade dos cuidados dos idosos e de todos que necessitem de assistência social. Através do desenvolvimento de interfaces e do uso de sensores capazes de capturar dados relativos a parâmetros físicos, cognitivos e funcionais, será possível monitorizar o utente. Ao mesmo tempo, a informação obtida permitirá também aos cuidadores fazer uma avaliação do estado de saúde do idoso e determinar a necessidade de um plano de intervenção específico.

Com base neste projeto, o objetivo do estágio passa por desenvolver um módulo de processamento de dados de sensores físicos. Este deverá permitir identificar padrões de atividade com base nos dados recebidos dos sensores instalados, lançar notificações nos casos em que é observada uma mudança no comportamento normal do utilizador, e por fim usar dados de múltiplos sensores e lógica temporal para detetar eventos mais complexos. A publicação de um artigo sobre o simulador desenvolvido na conferência HCist 2018 (*International Conference on Health and Social Care Information Systems and Technologies*) é mais um sinal do enorme potencial de todo o trabalho realizado.

Palavras-Chave

Envelhecimento, Cuidados Continuados, Internet das Coisas, Monitorização, Processamento de dados, Identificação de Padrões, Sensores.

Abstract

Nowadays, the population's aging is one of the major challenges facing the world. The increase in the average life expectancy is one of the main causes of the demographic and socioeconomic changes and, consequently, of the increase of elderly people. Therefore, it is necessary to pay more attention to this age group, in order to reduce the risk of social exclusion.

Due to the lack of technological solutions that ensure interoperability between social institutions, it is necessary to develop information systems in order to improve communication between the actors. The generalization of the IoT solutions and the enhancement of the integrating capacities of the systems can guarantee a closer proximity to citizen monitoring and evaluation.

In order to overcome these problems, the SOCIAL platform (Social Cooperation for Integrated Assisted Living), aims to improve the integration's care of the elderly and those in need of social assistance. Through the development of interfaces and the use of sensors capable to capture data related to physical, cognitive and functional parameters, it will be possible to monitor the user. At the same time, the information obtained will also enable caregivers to assess the health status of the elderly and determine the need for a specific intervention plan.

Based on this project, the goal of the internship is to develop a module for processing physical sensor data. This should enable the identification of activity patterns based on data received from the installed sensors, launch notifications, in cases where a change in the normal behaviour of the user is observed, and finally use data of multiple sensors and temporal logic to detect more complex events. The publication of an article about developed simulator at the HCIST 2018 (International Conference on Health and Social Information Systems and Technologies) conference is another indication of the enormous potential of all the work that has been done.

Keywords

Aging, Continuing Care, Internet of Things, Monitoring, Data Processing, Pattern Recognition, Sensors.

Índice

Lista de Figuras xi

Lista de Tabelas xiii

Lista de Acrónimos xv

Capítulo 1 Introdução 1

1.1 Descrição do problema 1

1.2 Descrição do projeto 3

1.3 Objetivos da Componente de deteção de eventos significativos 5

1.4 Estrutura do documento 6

Capítulo 2 Estado da arte 9

2.1 Principais características 9

2.2 Tecnologias AAL 11

2.3 Resumo da pesquisa realizada 17

Capítulo 3 Requisitos e Atributos de Qualidade 19

3.1 Personas e Cenários 20

3.2 Requisitos funcionais 25

3.2.1 Atores 25

3.2.2 Casos de uso 25

3.3 Atributos de Qualidade 28

3.4 Restrições 30

Capítulo 4 Estrutura do projeto e experiências preliminares 33

4.1 Visão geral 33

4.2 Descrição da Componente de deteção de eventos significativos 35

4.3 Seleção das tecnologias 35

4.3.1 Ferramenta para processamento de dados em tempo real 36

4.3.2 Protocolo de comunicação 36
4.3.3 Protocolo de mensagens 36
4.3.4 Base de dados 37
4.4 Persistência dos dados em FHIR 37
4.5 Segurança e privacidade 38
4.6 Experiências preliminares 39
Capítulo 5 Arquitetura 45
5.1 Arquitetura da componente deteção de eventos significativos 46
5.1.1 Máquinas de estados 47
5.2.2 Filtros 53
Capítulo 6 Implementação 55
6.1 Gerador de eventos 55
6.2 Processamento dos dados 57
6.3 Execução do sistema 58
Capítulo 7 Testes e Resultados 61
7.1 Detetar estado de descanso com sensor de cama 61
7.2 Detetar estado de descanso com sensor de cama e de janela 62
7.3 Detetar entrada e saída de casa 63
7.4 Testes funcionais 65
Capítulo 8 Gestão do trabalho 67
8.1 Metodologia 67
8.2 Planeamento 69
8.2.1 Primeiro Semestre 69
8.2.2 Segundo semestre 71
8.3 Riscos 73
Capítulo 9 Conclusão 77
Referências 79
Anexos 83

Anexo A Artigo sobre a componente desenvolvida..... 85

Anexo B Resultados dos testes às máquinas de estados..... 91

Lista de Figuras

Figura 1.1 – Diagrama de casos de uso da aplicação web "*Social Operation Management*".....4

Figura 1.2 – Diagrama de casos de uso da aplicação móvel "*My Social*".....4

Figura 2.1 - Visão e conceitos essenciais do projeto "inCASA"[15]. 13

Figura 2.2 - *Workflow* dos algoritmos usados no projeto "*Fit4Work*".[18].....13

Figura 2.3 - Descrição dos componentes do projeto "*Relaxed Care*".[19].....14

Figura 2.4 - Ilustração das interações presentes no projeto "CAMI".[21].....15

Figura 3.1 - Diagrama de casos de uso. 26

Figura 4.1 - Localização da Componente de deteção de eventos significativos na arquitetura da plataforma SOCIAL. 34

Figura 5.1 - Visão alto nível da arquitetura do sistema de deteção de eventos significativos. 46

Figura 5.2 – Máquina de estados para detetar se um utilizador está a descansar. 47

Figura 5.3 – Máquina de estados para detetar se um utilizador está a descansar com recurso a sensores na cama e na janela do quarto. 48

Figura 5.4 – Máquina de estados para detetar se um utilizador tem a televisão ligada. 49

Figura 5.5 – Máquina de estados para detetar se um utilizador está em casa. 50

Figura 5.6 – Exemplo de código Java relativo ao estado *Door Open*. 51

Figura 5.7 – Máquina de estados de alertas relacionados com o descanso do utilizador. 51

Figura 5.8 – Máquina de estados de alertas relacionados com a presença de utente em casa 52

Figura 5.9 – Máquina de estados de alertas relacionados com a presença de utente em casa e a televisão. 52

Figura 5.10 – Plano sequencial de filtros para detetar se o idoso está a descansar 54

Figura 6.1 – Exemplo de ficheiro de configuração de uma simulação 56

Figura 6.2 – Exemplo de ficheiro XML de configuração de filtros 57

Figura 6.3 – Página web com a listagem dos eventos do idoso 59

Figura 8.1 - Representação por fases do modelo ACDM.....68

Figura 8.2 - Diagrama de Gantt das tarefas executadas no primeiro semestre. 70

Figura 8.3 - Gráfico de distribuição do esforço semanal, por tarefa, no primeiro semestre. .71

Figura 8.4 - Diagrama de Gantt das tarefas planeadas para o segundo semestre.....71

Figura 8.5 - Diagrama de Gantt das tarefas executadas no segundo semestre. 72

Figura 8.6 - Gráfico de distribuição do esforço semanal, por tarefa, no segundo semestre...73

Lista de Tabelas

Tabela 3.1 - Descrição dos casos de uso identificados. 26

Tabela 3.2 - Descrição dos atributos de qualidade do projeto. 29

Tabela 3.3 - Descrição das restrições de negócio do projeto. 30

Tabela 3.4 - Descrição das restrições técnicas do projeto.....31

Tabela 4.1 - Experiência com Spark /Spark Streaming. 41

Tabela 4.2 - Experiência com *Message Queues*. 42

Tabela 4.3 - Experiência com um sensor de presença (iTag). 43

Tabela 4.4 - Utilização de dados no formato standard (XMPP). 44

Tabela 7.1 – Resultados da execução para deteção do estado de descanso de um utilizador 62

Tabela 7.2 – Resultados da execução para deteção do estado de descanso de um utilizador, com dados de um sensor de contacto na janela e um de pressão na cama durante 10 dias ...63

Tabela 7.3 – Resultados da execução para deteção de saída e entrada em casa de um utente. 64

Tabela 7.4 – Resultados da execução da simulação para deteção de estado de descanso e de eventos de saída e entrada em casa de um utente. 65

Tabela 7.5 – Resultados dos testes funcionais do sistema. 66

Tabela 8.1 - Tabela de identificação dos riscos do projeto. 74

Lista de Acrónimos

AAL - Active and Assisted Living

ABAC - Attribute-Based Access Control

ACDM - Architecture Centric Design Model

ADSL - Asymmetric Digital Subscriber Line

DEI - Departamento de Engenharia Informática

FHIR - Fast Healthcare Interoperability Resources

HCist - International Conference on Health and Social Care Information Systems and Technologies

HTML - Hypertext Markup Language

IoT - Internet of Things

IPN-LIS - Instituto Pedro Nunes - Laboratório de Informática e Sistemas

IPSS - Instituições Particulares de Solidariedade Social

JSON - JavaScript Object Notation

LTSA - Labelled Transition System Analyser

MQTT - Message Queuing Telemetry Transport

NFC - Near Field Communication

NoSQL - Not Only Structured Query Language

PEP - Policy Enforcement Point

RBAC - Role-Based Access Control

RDD - Resilient Distributed Datasets

SOCIAL - Social Cooperation for Integrated Assisted Living

SQL - Structured Query Language

TCP - Transmission Control Protocol

TRL – Technology Readiness Levels

XACML - eXtensible Access Control Markup Language

XML - Extensible Markup Language

XMPP - Extensible Messaging and Presence Protocol

Capítulo 1
Introdução

Segundo previsões do Instituto Nacional de Estatística, em 2080, o número de pessoas com mais de 65 anos a viver em Portugal passará dos 2.1 para os 2.8 milhões de habitantes [1]. Porém, a população total não irá acompanhar este crescimento, passando dos atuais 10.3 milhões de habitantes para os 7.5 no mesmo período. O aumento da esperança média de vida à nascença e a diminuição da taxa de natalidade são fatores que contribuem para que, no futuro, o número de idosos corresponda a mais de um terço da população total do país.

Tendo em conta esta realidade, torna-se essencial proporcionar boas condições para que os últimos anos de vida da população sénior sejam de qualidade, sendo fundamental a integração e continuidade dos cuidados. Para isso ser possível, é necessário desenvolver soluções capazes de responder às diferentes realidades e que sejam capazes de se ajustar às mudanças constantes da sociedade, encarando a hospitalização e a institucionalização como um último recurso.

1.1 Descrição do problema

A rede de cuidados continuados em Portugal apresenta uma diversidade de contextos que permitem dar apoio ao idoso. Estes contextos dependem do tipo de cuidados prestados ou do nível de autonomia do utente, podendo ser divididos em três grupos. O primeiro consiste na rede de cuidados informais, da qual fazem parte familiares, amigos e vizinhos, bem como muitas organizações sociais, como grupos de voluntários, instituições religiosas ou organizações não-governamentais. Estes cuidadores também fazem parte de uma rede de suporte à integração e continuidade de cuidados. O segundo grupo engloba a rede de cuidados formais para idosos autónomos, do qual fazem parte atividades como a Universidade sénior, os passeios, a atividade física ou a hidroginástica. Por fim, fazem parte do último grupo os idosos não autónomos que não são capazes de realizar determinadas tarefas sem o auxílio de um cuidador. Mesmo nesta categoria de cuidados, o nível de dependência vai desde um grau mais baixo, com a preparação de medicamentos, o apoio domiciliário ou com a entrega de refeições, até um grau mais elevado, como o acompanhamento de um utente num lar de idosos.

A cooperação entre os prestadores de cuidados é algo fundamental para garantir a qualidade de um serviço. Porém, isto só se torna possível através de mecanismos eficientes para a comunicação e partilha de informação, algo que atualmente não é disponibilizado. Na maioria dos casos, a informação está guardada em formatos diferentes e dispersa por cada uma das bases de dados das entidades. Para além disso, existe alguma relutância em partilhar esses dados com outras entidades, como as câmaras municipais, dando lugar a dados duplicados sobre a mesma pessoa. Por exemplo, caso seja necessário contactar algum familiar em caso de emergência, algumas instituições poderão não ser capazes de o fazer porque a informação existente sobre a população idosa será insuficiente.

Os problemas enumerados anteriormente vão ter consequências na prestação de cuidados e na falta de apoio aos idosos por parte de todas as entidades envolvidas. A integração dos mesmos irá sofrer algumas complicações associadas à fragmentação dos serviços, que não dão garantias dos serviços mínimos necessários. Além disso, a resposta dada a imprevistos, como um internamento hospitalar urgente, não será tão eficaz, porque a informação disponível sobre o histórico do utente poderá ser insuficiente para um tratamento adequado, podendo levar a consequências mais graves.

Tendo em conta o âmbito da prestação de cuidados, é necessário melhorar as soluções tenológicas, desenvolvendo interfaces que se adequem às características dos diferentes atores, bem como aos dispositivos que as utilizam. Por exemplo, se estivermos a falar de uma aplicação direcionada à população mais idosa, o *design* da mesma terá que ser intuitiva, com ícones auxiliares à navegação, e o tamanho de letra utilizado deverá ser grande, devido aos problemas de visão que afetam esta faixa etária [2]. As soluções tecnológicas aliadas a uma melhoria da interoperabilidade, iriam contribuir para uma melhoria das comunicações, permitindo às várias entidades trocarem e usarem informações da população entre si.

A utilização de sensores também é vista como uma necessidade, porque permitirá garantir um melhor acompanhamento do idoso. A informação aqui recolhida poderá servir para a criação de planos de intervenção, mapas de referenciação geográfica ou constituição de programas e atividades direcionadas para esta faixa etária. Porém, deverá ser garantida a privacidade e proteção dos dados, para que estes não sejam partilhados de forma indevida.

1.2 Descrição do projeto

A plataforma SOCIAL (http://www.kentratech.eu/social/) é um projeto inserido no programa "Portugal 2020", iniciado em Julho de 2016, que pretende construir uma infraestrutura de suporte ao registo de dados associados à esfera da saúde e social de um cidadão. O principal objetivo passa por melhorar a prestação de cuidados aos indivíduos e à comunidade, fazendo o devido acompanhamento ao longo dos seus ciclos de vida e fazendo a ponte entre estes dois domínios.

A partir das necessidades que foram detetadas, após a realização das entrevistas às instituições interessadas (câmaras municipais, IPSS) e aos utentes alvo, foram definidos os seguintes objetivos para este projeto:

- Colmatar as lacunas em termos de partilha de informação, fornecendo um conjunto de componentes estruturais que garantam uma melhor comunicação entre os vários atores da rede de cuidados.
- Disponibilizar mecanismos eficientes para a persistência de informação.
- Garantir a interoperabilidade com a rede de cuidados formais, como os centros de dia, hospitais ou farmácias.
- Ajudar decisores na criação de programas direcionados para os idosos, com base na utilização secundária da informação recolhida.
- Desenvolver interfaces e usar sensores capazes de monitorizar o utente, permitindo fazer uma avaliação da situação atual e criar um plano de intervenção específico, caso seja necessário.

Para este projeto serão desenvolvidas uma aplicação web ("*Social Operation Management*"), que permitirá aos Utilizadores, Profissionais e Cuidadores gerir grande parte das componentes da plataforma social relacionadas com utentes, pedidos de assistência, notificações, entre outros, e uma aplicação móvel ("*My Social*") que permitirá que uma pessoa se registe e tenha acesso a um conjunto de serviços que a plataforma disponibiliza. Os diagramas de casos de uso das mesmas foram definidos na fase inicial do projeto por todos os elementos do consórcio e encontram-se representados nas Figuras 1.1 e 1.2, respetivamente.

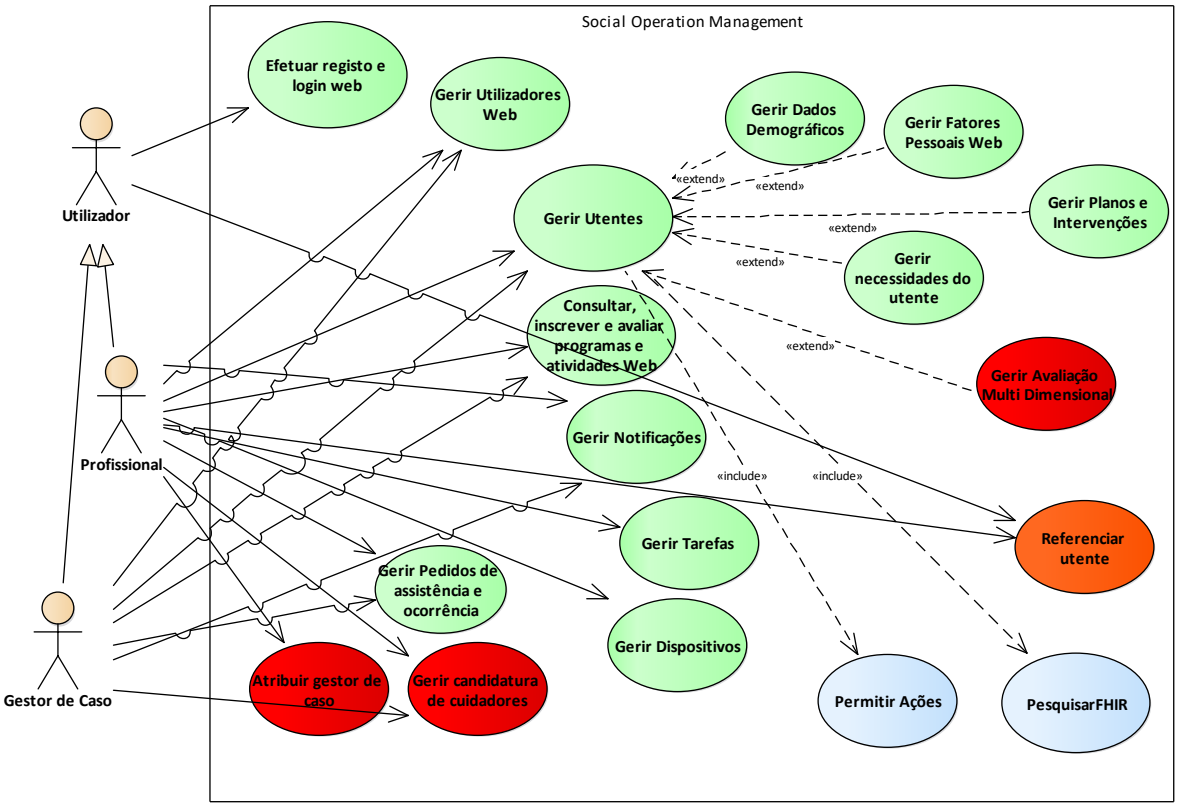

Figura 1.1 – Diagrama de casos de uso da aplicação web "*Social Operation Management*"

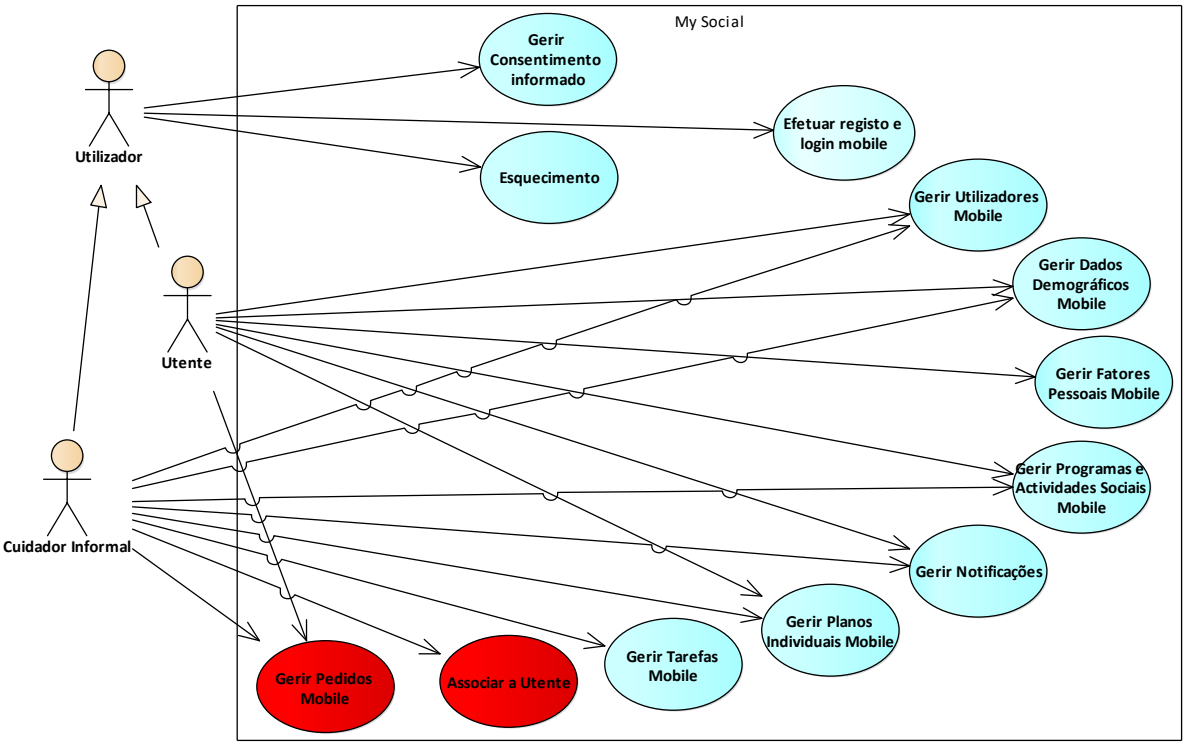

Figura 1.2 – Diagrama de casos de uso da aplicação móvel "*My Social*"

Deste projeto fazem parte cinco parceiros com conhecimentos e experiência comprovada que vão contribuir para o desenvolvimento do mesmo, sendo eles:

- **Kentra Technologies, Lda** - Promotor líder do consórcio que apresenta soluções na área da saúde já no mercado. Irá contribuir para o projeto com recursos especializados na área de desenvolvimento de *software*, bem como experiência no mercado da saúde.
- **Digitalwind** - Empresa que fornece soluções na área das tecnologias da informação e apresenta a experiência e os recursos na área de desenvolvimento de Sistemas de Informação como suporte.
- **Universidade de Aveiro** - Estabelecimento público de ensino superior que contará com uma equipa de investigadores com fortes antecedentes tecnológicos e conhecimentos valiosos na área da saúde.
- **Micro I/O** - Empresa especializada em soluções integradas de *hardware* e *software*, contribuindo para o projeto através da integração de sensores e de sistemas de gestão de controlo. Especificamente, contribuirá com o desenvolvimento de soluções e interfaces de suporte à interação do utilizador com a plataforma, para acompanhamento e monitorização de atividade, bem como comunicação das leituras dos equipamentos à plataforma, para armazenamento e análise.
- **Instituto Pedro Nunes - Laboratório de Informática e Sistemas** - Instituição privada sem fins lucrativos que apresenta vasta experiência no desenvolvimento de plataformas e Sistemas de Informação. Contribuirá na definição da solução global e integrará os grupos de desenvolvimento, testes e piloto.

1.3 Objetivos da Componente de deteção de eventos significativos

Dos planos da plataforma SOCIAL faz parte o desenvolvimento de uma componente capaz de, através da instalação de sensores nas habitações da população idosa, recolher dados físicos ou funcionais sobre o utente. Desta forma, será possível fazer uma avaliação das condições existentes e criar um plano de intervenção específico, caso seja necessário. Para que tal seja possível é necessário desenvolver esta componente e testá-la antes de a integrar em ambiente real.

Posto isto, o objetivo principal para este estágio passou por desenvolver um protótipo de um módulo de processamento de dados de sensores fisicos e testá-lo. Pretendia-se atingir um nível de maturidade tecnológica de nível 4 (*Technology Readiness Levels* - TRL 4), ou seja,

fazer uma validação do componente em ambiente laboratorial. Para tal, era necessário ter acesso a um grande conjunto de dados de sensores. Porém, esta abordagem pode apresentar limitações em termos de custo dos sensores, disponibilidade e tempo de desenvolvimento [3]. Para resolver esse problema foi definida a implementação de um simulador que permitisse gerar dados, previamente configurados para cada casa/pessoa. Não existindo ainda sensores instalados nas casas dos idosos, a utilização de um ambiente simulado poderá ajudar a atenuar esses problemas e facilitar a geração de um conjunto de dados fiável para testes ao sistema.

A Componente de deteção de eventos deve permitir:

- Identificar padrões de atividade com base nos dados recebidos de um ou mais sensores, pretendendo-se, desta forma, detetar ações simples como deitar e levantar da cama ou entrar e sair de casa.
- Detetar casos onde há um desvio relativamente ao comportamento normal do utilizador. Usando os exemplos do ponto anterior, estes eventos poderão ser levantar ou deitar muito antes do normal para um determinado utente ou sair de casa a meio da noite sem razão aparente.
- Detetar eventos mais complexos, usando múltiplos sensores e lógica temporal. Para o caso de detetar entradas e saídas pela porta principal, será necessário usar pelo menos dois tipos de sensores e associar a ordem de receção dos dados com o evento correspondente.

Pretende-se que o trabalho realizado ajude a perceber, através dos dados recebidos, eventos realizados no quotidiano dos utentes, lançando notificações aos cuidadores no sentido de lhes proporcionar uma melhor qualidade nos cuidados prestados, como por exemplo, a que horas o utilizador abriu as janelas de casa ou quando é que o utilizador ligou a televisão. Caso se verifique um comportamento anormal, o cuidador deverá ser notificado sobre essa situação.

1.4 Estrutura do documento

Para além da introdução, este documento apresenta mais oito capítulos, que se dividem da seguinte forma:

- **Estado da arte** – Neste capítulo será feita uma análise aos vários processos voltados para o tratamento de dados e serão analisados vários projetos que apresentam características semelhantes ao que irá ser implementado.
- **Requisitos e Atributos de qualidade** - Neste capítulo serão descritos os requisitos do trabalho, de forma mais detalhada, bem como os atributos de qualidade do projeto e restrições.
- **Estrutura do trabalho e Experiências preliminares** – Será descrita a estrutura do trabalho, bem como as interações existentes com os outros componentes. Além disso serão apresentadas algumas experiências preliminares realizadas à componente a desenvolver.
- **Arquitetura** - As diferentes vistas das várias componentes do projeto serão apresentadas neste capítulo, assim como as interações existentes entre as mesmas.
- **Implementação** – Este capítulo apresentará todos os passos dados durante a implementação do projeto.
- **Testes e Resultados** – Neste capítulo serão descritos os resultados obtidos após a implementação do sistema.
- **Gestão do trabalho** - Esta secção do documento apresentará todos os pontos essenciais para a gestão do trabalho, tais como o planeamento das tarefas e a gestão dos riscos.
- **Conclusão** - No último capítulo será feita uma retrospetiva acerca dos resultados obtidos e apresentar-se-ão as perspetivas futuras/objetivos futuros a realizar no contexto de continuação deste trabalho.

Capítulo 2
Estado da arte

O Estado da arte é essencial para compreender o trabalho científico e industrial realizado até ao momento, sendo uma das partes mais importantes de todo o projeto. Esta atividade de pesquisa permite fazer comparações com projetos já existentes, demonstrar que aquilo que se pretende desenvolver apresenta carácter de novidade e adquirir alguns conhecimentos que permitam a melhoria da ideia inicialmente proposta.

Neste capítulo serão descritas as características que compõem o sistema e sobre as quais foi necessário efetuar uma pesquisa, de forma a perceber como estariam a ser aplicadas noutros projetos e a viabilidade das mesmas no nosso. Por fim, será apresentada uma revisão bibliográfica acerca dos trabalhos realizados na mesma área e dos pontos comuns com o projeto a ser desenvolvido.

2.1 Principais características

Antes de avançar na pesquisa do Estado da arte, é necessário definir as características que permitem fazer essa análise. Com base nas necessidades reveladas nas entrevistas realizadas numa fase anterior ao início deste estágio e em alguns atributos do sistema, foram selecionados os pontos que definem as principais características da Componente de deteção de eventos do projeto SOCIAL e que estarão relacionadas principalmente com o tratamento dos dados. Estas são descritas nos próximos pontos:

- **Deteção de atividade na habitação a partir de dados de sensores simples** - Consta como uma das necessidades da plataforma social, porque é fundamental para fazer o acompanhamento do idoso quando não se encontra nenhum cuidador por perto para ajudar. Os sensores produzidos pela empresa responsável apresentam custos de produção mais baixos do que o habitual, sendo uma das vantagens deste projeto. Tapia e Intille [4] mostram que a partir de um pequeno conjunto de dados de sensores instalados em diversas habitações é possível reconhecer diversas atividades de interesse para a área médica, como a higiene do utente, com taxas de sucesso entre os 25% e os 89%, dependendo do critério utilizado. Também Skocir *et all* [5], demonstraram o uso de sensores simples para a deteção de atividades do dia-

a-dia como abrir ou fechar a porta, no entanto aplica diferentes algoritmos para a deteção desses eventos.

- **Monitorização não invasiva** - Nas entrevistas realizadas percebeu-se que os utentes concordavam com a monitorização, desde que não se pusesse em causa a sua privacidade. Nesse caso deverá ser evitado o uso de um *smartwatch* para detetar o estado atual do idoso [6] ou câmaras, sensores de localização e microfones para controlar o ambiente [7].
- **Envio de alertas de eventos significativos** - Este ponto é especialmente importante para os cuidadores acompanharem os utentes. Caso seja detetado um evento que não seja normal, os cuidadores são informados e podem intervir.
- **Identificação de padrões** - Capacidade de deteção de rotinas de um utente, para que o sistema seja capaz de avaliar todos os eventos recebidos e detetar aqueles que poderão ser significativos. Sebestyen et al. [8] utilizaram modelos ocultos de Markov para o reconhecimento de atividade dos idosos dentro das suas casas. Desta forma, foi possível determinar as atividades de uma pessoa, num determinado local, porque apesar de o ser humano agir naturalmente e não segundo regras, como um robot, mantém algumas rotinas e isso verifica-se principalmente com os idosos. Moraru et al. [9] usaram dados não tratados de sensores para extrair informações úteis e aprender algumas rotinas através de algoritmos de *machine learning*. Para o conjunto de dados utilizados, as árvores de decisão e as redes Bayesianas foram as que apresentaram melhores resultados comparativamente aos algoritmos de regressão linear. Porém, para tirar conclusões gerais eram necessárias mais experiências e um *dataset* mais extenso. Além disso, concluíram também que o modelo gerado pelo algoritmo de árvores de decisão era o mais fácil de interpretar apresentando também bom desempenho.
- **Qualidade dos dados** – Os dados recolhidos por parte dos sensores servirão como base para o reconhecimento de atividade por parte do idoso. Se estes não apresentarem qualidade suficiente, as decisões tomadas pelo sistema podem não ser as corretas e, por isso, este ponto é essencial para o funcionamento dos sistemas de informação [10]. Karkouch et al. [11] destacaram a importância da qualidade dos dados e enumeram alguns problemas que podem afetar o resultado final, como a escala de desenvolvimento, devido à grande quantidade e variedade da informação recebida, a rede, o tipo de sensores utilizados ou até o ambiente em que estes são instalados. Problemas como a falta de dados, a sua duplicação ou o envio incorreto

dos mesmos são apontados como os mais frequentes. Kara et al. [12] apresentaram fatores para a qualidade dos dados relacionados com os sistemas AAL (*Ambient Assisted Living*). A informação recolhida deverá corresponder o máximo possível à realidade, os dados deverão ser completos e todo o sistema necessita de ser eficiente, preciso, confiável e seguro.

- **Processamento dos dados** – Refere-se ao conjunto de tarefas que transformam os dados recebidos dos sensores em informação útil para a plataforma. Em primeiro lugar, é necessário fazer uma limpeza dos dados de forma a evitar *outliers*. Os *outliers* são dados que são recebidos fora daquilo que é considerado como "estado normal" e são uma das principais manifestações da incerteza dos dados [11]. De forma a evitar esse problema é possível aplicar métodos estatísticos, como a média, o desvio padrão ou algoritmos de *clustering* [13]. A segunda fase do processamento passará por analisar nos dados tratados e detetar eventos. Skocir et al. [5] apresentaram duas abordagens para a deteção de atividade em casas com sensores instalados: uma com recurso a uma janela deslizante e outra que utiliza redes neuronais. Após comparar cada um dos métodos com os eventos que realmente ocorreram, concluiu-se que a primeira abordagem apresentava uma taxa de precisão de 60% e que a segunda apresentava uma taxa inferior de acerto, de cerca de 30%. Noutro projeto, de forma a detetar atividades como tomar banho, sair de casa ou ir para a cama, Hong e Nugent [14] proposeram três abordagens para o processamento dos dados: uma baseada na localização dos sensores, outra baseada em modelos e outra também em modelos, mas que apresenta um sensor dominante. Todas as abordagens apresentarem taxas de acerto semelhantes, na ordem dos 80%.

2.2 Tecnologias AAL

As tecnologias AAL correspondem a produtos tecnológicos inovadores que promovem o envelhecimento ativo e saudável. Nesta secção do capítulo do estado da arte serão apresentados alguns projetos que se inserem na mesma área do projeto SOCIAL. Será feita uma breve descrição dos mesmos sobre o contexto e forma de funcionamento. Tendo como objetivo a retenção de alguns conceitos relativos a envelhecimento assistido e sensores, pretende-se principalmente entender algumas das técnicas usadas no desenvolvimento destes sistemas, se possível.

InCasa Project

O projeto "*inCASA*" [15] é um projeto criado pela Comissão Europeia que pretende criar tecnologias dirigidas a cidadãos com mais de 65 anos e uma rede de serviços que proteja os idosos, prolongando o tempo de permanência na sua própria casa com qualidade.

Este projeto tem como principal inovação a presença de um módulo inteligente capaz de criar um perfil físico e comportamental dos pacientes idosos. Os sensores irão monitorizar parâmetros clínicos, como o número de batimentos cardíacos, o peso ou o nível de pressão arterial e os hábitos do utilizador, bem como os movimentos dentro da sua casa, a presença na cama, detetar se a porta da frente está aberta ou fechada ou se a televisão está ligada ou não. Os dados recebidos dos sensores são guardados numa base de dados disponível a todas as entidades. Além disso, a partir desta plataforma é possível enviar alertas para os cuidadores sempre que se verifique alguma alteração no comportamento do idoso. A Figura 2.1 faz referência ao que foi enumerado anteriormente.

Os projetos piloto realizados [16] apresentaram resultados que indicam que a integração de serviços pode melhorar a qualidade de vida dos pacientes mais frágeis e vulneráveis, melhorando os resultados clínicos e direcionando os cuidados de forma mais eficaz. No entanto, as barreiras culturais, relativas à partilha de dados entre os vários elementos que compõem a assistência social, mostram que ainda há muito a ser melhorado neste ramo. Essa mudança só se tornará possível com uma forte intervenção dos legisladores e uma grande mudança cultural.

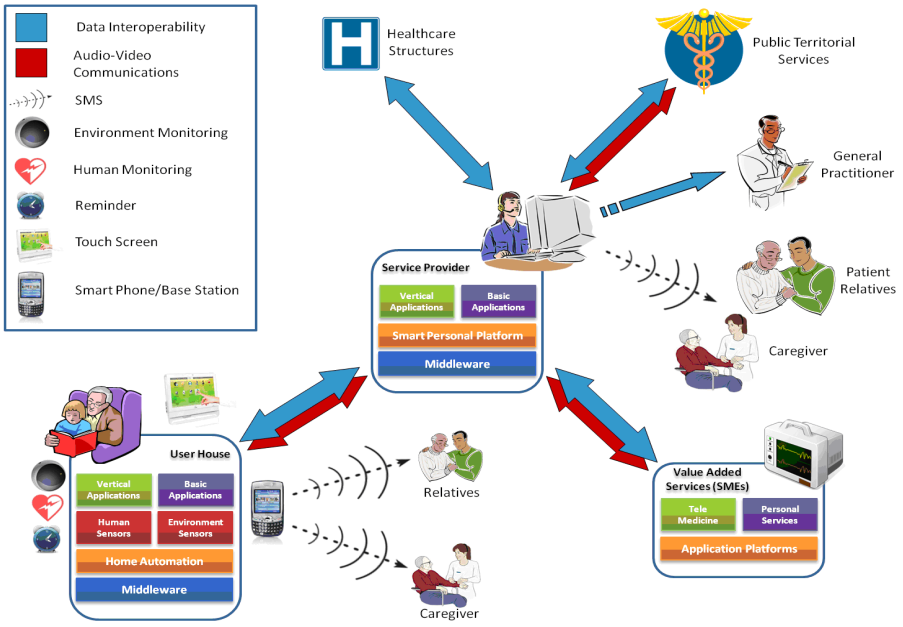

Figura 2.1 - Visão e conceitos essenciais do projeto "inCASA"[15].

Fit4work

O projeto "*Fit4Work*" [17] faz parte do *AAL programme* (www.aal-europe.eu/) e tem como objetivo desenvolver um sistema inovador, de simples utilização e discreto que apoiará os trabalhadores com mais de 55 anos na redução e gestão de problemas resultantes da sua profissão. A partir dos dados recolhidos pelos sensores presentes nos *smartphones* e/ou *wristbands*, como o acelerómetro ou o sensor de frequência cardíaca, e dos sensores ambientais, como a temperatura, humidade e concentração de dióxido de carbono, combinados com algoritmos de aprendizagem computacional, é possível calcular o nível de *stress* físico, mental e ambiental de um indivíduo, baseando-se em evidências científicas.

Com base nestes valores, o sistema vai fornecer recomendações adequadas a cada utilizador e definir planos de estilo de vida a médio e longo prazo, tendo em conta o tipo de trabalho e a idade do indivíduo. Desta forma, pretende-se reduzir o *stress* mental, manter um nível adequado de atividade física e melhorar as condições de trabalho.

Na Figura 2.2, é possível observar o fluxo de trabalho deste algoritmo. No lado esquerdo é notório o pré-processamento, a normalização de dados e a localização do contexto. O lado superior direito corresponde ao módulo de acompanhamento de atividade e a parte inferior direita ao módulo de acompanhamento dos níveis de *stress*. [18]

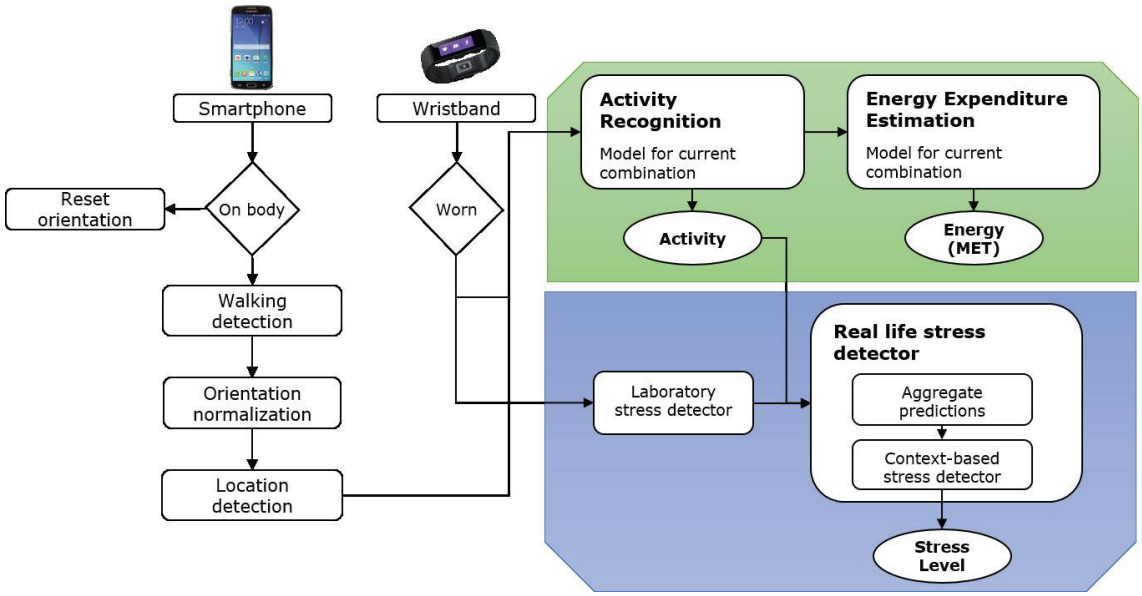

Figura 2.2 - *Workflow* dos algoritmos usados no projeto "*Fit4Work*".[18]

Relaxed Care

O projeto "*Relaxed Care*", coordenado pelo Instituto Austríaco de Tecnologia [19], apresenta um paradigma de comunicação simples concebido para o envio de mensagens básicas entre os idosos e os seus cuidadores. Este sistema é constituído por um cubo, que do lado do idoso está colocado numa localização central da casa, conectado a vários sensores. Do lado do cuidador, este cubo poderá ser colocado em casa ou no trabalho. Desta forma, será possível monitorizar o estado atual de um idoso, os seus níveis de atividade, interação social e as suas rotinas diárias. Além disso, a partir deste cubo é possível saber o estado de espírito a partir de um indicador de cor existente na parte frontal do mesmo.

A comunicação entre utilizadores é feita através de pequenas esferas, cada uma associada a uma função diferente. Estas são colocadas na parte superior do cubo e serão identificadas a partir de sensores NFC (*Near Field Communication*), tal como se pode observar na Figura 2.3. O *design* apelativo desta solução permite uma comunicação simples e de fácil utilização para os idosos.

Existe ainda uma aplicação que tem as mesmas funcionalidades deste cubo mas que, além disso, permite consultar informação detalhada sobre o histórico de comunicação entre idoso e cuidador.

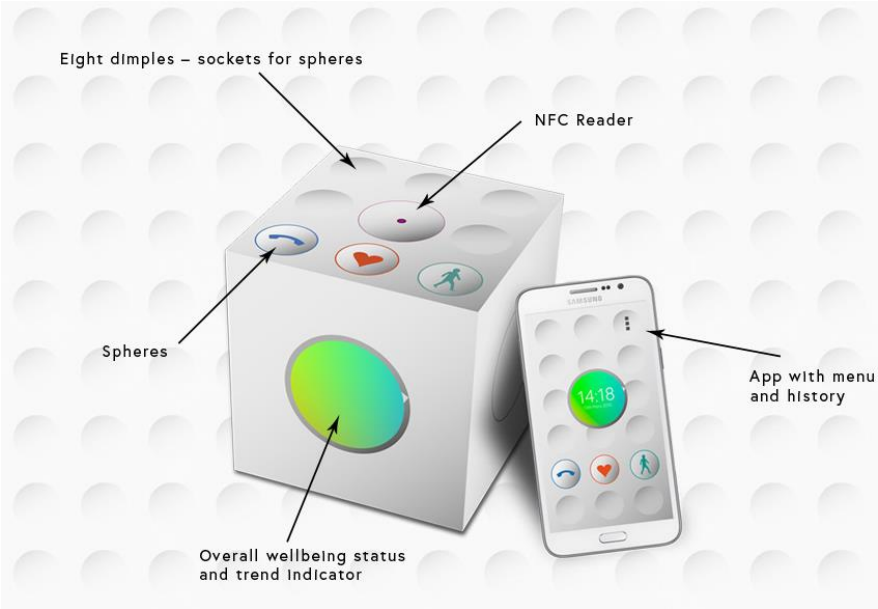

Figura 2.3 - Descrição dos componentes do projeto "*Relaxed Care*".[19]

CAMI project

Inserido no contexto do envelhecimento, foi criado o projeto CAMI [8, 9], que também faz parte do *AAL Programme*. Para esta solução foram integradas várias tecnologias, como a telepresença robótica, a integração de sensores portáteis e dispositivos para monitorizar o estado de saúde e deteção de quedas, bem como dispositivos domésticos inteligentes para integração de sensores e eletrodomésticos de forma a controlar o ambiente vital. A Figura 2.4 ilustra as interações presentes no sistema.

O objetivo deste projeto passa por dar resposta ao aumento da procura de cuidados observado atualmente e é dirigido a idosos saudáveis, assim como àqueles que apresentam algumas limitações. Desta forma, pretende-se que seja retirada alguma carga de esforço aos familiares que também são cuidadores destas pessoas.

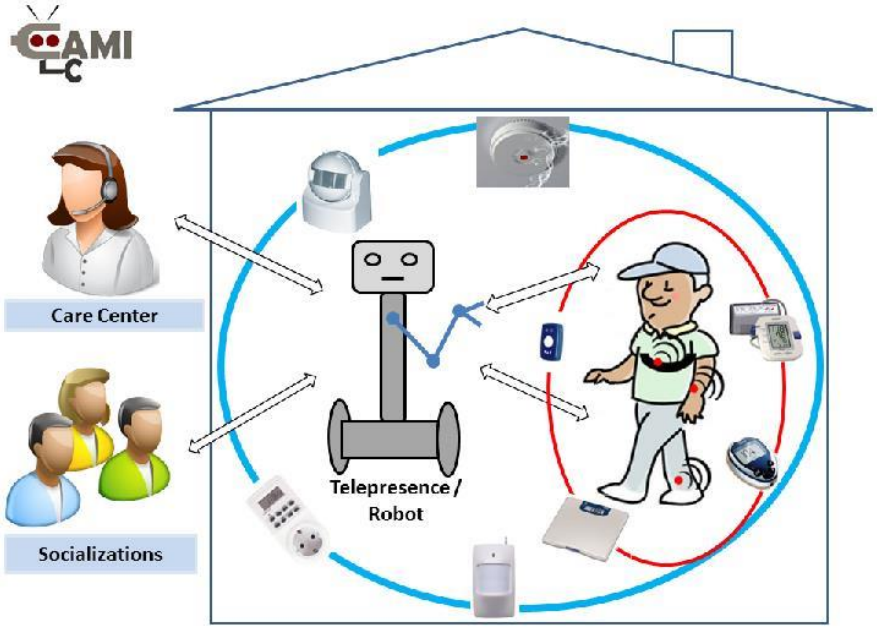

Figura 2.4 - Ilustração das interações presentes no projeto "CAMI".[21]

eCAALYX

O projeto eCAALYX [22], criado ao abrigo do *AAL Programme* e coordenado pela *Fundació Privada CETEMMSA,* foi criado com o intuito de melhorar a vida da população idosa, promovendo a sua autonomia e qualidade de vida e permitindo reduzir os custos de hospitalização. Está especificamente direcionado para questões relacionadas com as doenças crónicas que afetam esta faixa etária, como são os casos das doenças cardiovasculares, doença pulmonar obstrutiva crónica ou diabetes tipo II.

Através da utilização da televisão de forma interativa, o eCAALYX permite aos seus utilizadores monitorizar a sua própria saúde, receber informações práticas sobre o seu estado de saúde e comunicar com os seus cuidadores. Para recolher dados sobre o idoso, como o peso ou a pressão arterial, são usados sensores Bluetooth, que vão comunicar com um *router* que os processará e fará o envio para um servidor central.

Os autores deste projeto procuraram criar interfaces que fossem compreensivas e que tivessem as características de apresentação e de texto adequadas aos seus utilizadores. Após a realização de testes com o público-alvo [23], foram encontrados alguns pontos que eram necessários aplicar para uma experiência com maior qualidade por parte dos idosos. Alguns exemplos desses pontos passam por minimizar o número de passos para cumprir uma determinada tarefa, reduzir a informação apresentada para que os utilizadores possam focar a sua atenção num conceito de cada vez, mostrar claramente a secção e localização atuais, usar esquema de cores com alto contraste, usar linguagem simples ou dar tempo aos utilizadores para lerem toda a informação.

Health@Home

Ao abrigo do *AAL Programme,* o projeto Health@Home [12, 13] visa resolver problemas sociais relacionados com a prestação de serviços de saúde para idosos afetados pela insuficiência cardíaca crónica. Esta plataforma permite monitorizar os principais parâmetros físicos dos utilizadores, como o peso, pressão arterial, respiração ou postura, a partir de sensores instalados nas suas casas e de outros dispositivos sem fios.

O sistema é composto por uma arquitetura cliente/servidor. O lado do cliente inclui a casa do paciente que contém um *gateway* doméstico, todos os recursos de computação e comunicação necessários e sensores que farão a sua comunicação através de Bluetooth. Já o lado do servidor, composto pelas instalações do serviço de saúde, vai receber e processar os dados enviados pelo cliente, deixando os dados disponíveis no seu sistema de informação. A comunicação entre *gateway* e servidor é feita através de ADSL, sendo a banda larga móvel usada como canal secundário.

Com este modelo é possível planear as atividades realizadas pelos pacientes e permite que os cuidadores ou equipas médicas acompanhem à distância as diversas situações, através de mensagens enviadas para os seus telemóveis. Com base nessas informações, diminuir-se-á o tempo de resposta a emergências e poderá evitar-se o tratamento prolongado de um doente no hospital. Desta forma, o paciente irá observar uma melhoria da qualidade de vida e haverá uma redução de custos por parte do sistema nacional de saúde.

Os resultados deste projeto mostraram que os pacientes estiveram ativos, principalmente nos primeiros dias, e que o número de falsos positivos foi baixo, não tendo sido observados quaisquer problemas de conectividade, transmissão ou dados perdidos. A aceitação por parte dos pacientes, bem como dos médicos, foi bastante positiva.

2.3 Resumo da pesquisa realizada

A primeira parte da análise ao estado da arte foi importante para fazer uma divisão de mais alto nível de algumas das principais características que farão parte da ferramenta de deteção de eventos significativos. A pesquisa efetuada por cada um desses pontos permitiu conhecer melhor alguns dos procedimentos usados e os resultados obtidos pelas diferentes técnicas. Alguns dos algoritmos acima descritos podem funcionar como soluções para este projeto e serem aplicados para resolver problemas relativos ao processamento de dados e à identificação de padrões. Além disso, outros trabalhos já publicados vieram comprovar que os pontos principais definidos para o projeto são essenciais para que este sistema seja aceite pelos utilizadores e, consequentemente, tenha sucesso no mercado.

Na segunda parte deste capítulo, procedeu-se a uma avaliação de projetos inseridos na mesma área da componente do projeto SOCIAL desenvolvida neste trabalho. Em todos os projetos apresentados são identificadas duas características em comum: o foco do trabalho desenvolvido direcionado para os idosos e a utilização de sensores. Porém, em grande parte dos trabalhos desenvolvidos, os sensores utilizados são direcionados para a saúde, medindo, por exemplo, o batimento cardíaco ou a pressão arterial.

Dois pontos comuns a quase todos os projetos são o acompanhamento dos utentes por parte dos cuidadores e o envio de informações sobre os mesmos. Tento em conta que se pretende fazer um acompanhamento individualizado a quem mais precisa, estes pontos são fundamentais para que se consiga agir com rapidez.

Esta análise serviu para perceber quais são as tendências usadas no desenvolvimento de sistemas deste tipo, bem como as características que lhes estão associadas. Serviu também para mostrar que o trabalho realizado apresenta potencial de novidade na forma como trata os dados. O processamento dos dados e a identificação de padrões de atividade com base numa abordagem híbrida, onde são analisados os dados recebidos com o histórico guardado de eventos anteriores, permite que esta componente detete anomalias no comportamento dos utentes. Desta forma, o acompanhamento dos cuidadores poderá ser dado da mesma

maneira que era dada nos restantes projetos, não necessitando que esta seja tão invasiva para os utentes.

O sistema a desenvolver não obriga à criação de novos algoritmos para que este apresente fator de novidade. Pretende-se usar técnicas e tecnologias existentes e conjugar diferentes abordagens para a construção de uma ferramenta capaz de tratar dados de sensores.

Capítulo 3
Requisitos e Atributos de Qualidade

Analisadas as alternativas ao projeto que se pretende executar, o próximo passo passa por aprofundar o conhecimento do problema e identificar os requisitos da solução. No final desta análise pretende-se representar o sistema da melhor forma possível, facilitando, numa fase mais avançada, a procura de novas soluções.

O sistema que se pretende desenvolver tem como principal objetivo a monitorização dos utentes com maior necessidade de cuidados, através da instalação de sensores nas suas casas. Essa tarefa está a cargo de uma empresa especialista em *hardware*, que irá desenvolver e instalar diferentes tipos de sensores acordados com os restantes elementos do consórcio.

Tendo em conta o baixo custo de produção de sensores por parte desta empresa, os critérios de seleção basearam-se na utilização de *hardware* simples, capaz de detetar atividades da rotina dos idosos e gerar alertas para os cuidadores quando se detetam alterações nesses eventos. Isto não significa que o protótipo que se pretende desenvolver não esteja suscetível à adição de outros sensores. Porém, para o projeto SOCIAL foi definida a utilização dos seguintes tipos de sensores:

- **Contacto** - Usados em portas e janelas, detetam se estas estão abertas ou fechadas.
- **Movimento** - Instalados dentro das casas, detetam a presença de alguém naquele espaço.
- **Consumo** - Presentes nas tomadas da casa, detetam o consumo energético dos aparelhos ligados às mesmas.
- **Pressão** - Usados em camas e sofás, detetam se uma pessoa está deitada ou sentada.

Estes sensores vão recolher dados que serão enviados para uma *gateway* que, posteriormente, fará o envio para a componente desenvolvida. Fundamentado neste processo, os dados podem ser tratados e, através de uma análise híbrida com o histórico de dados do utente, será possível detetar eventos significativos, ou seja, ocorrências que não façam parte da normalidade e que podem necessitar da intervenção de um cuidador. A comunicação com este ator será efetuada através do envio de uma notificação/alerta no momento em que esse evento ocorrer.

3.1 Personas e Cenários

As personas correspondem à representação de grupos de potenciais utilizadores com características, necessidades, comportamentos, motivações e espectativas em comum [26]. Apesar de serem fictícias, as personas tem que ser precisas e descrever com exatidão a população alvo. Por outro lado, os cenários ajudam a perceber melhor o ambiente em que se insere uma persona e descrevem uma determinada situação. Os detalhes são importantes para representar os diferentes utilizadores e as circunstâncias em que o sistema a desenvolver poderá ser importante para corrigir determinadas situações.

De forma a suportar o processo de desenvolvimento tecnológico da plataforma SOCIAL e após a definição da metodologia para a criação de personas e cenários [26], foram criadas algumas personas do projeto com base em entrevistas realizadas a *stakeholders* relevantes e validadas, posteriormente, num *focus group*. As personas definidas para o projeto SOCIAL estão publicadas em artigo e dele fazem parte idosos ativos, sedentários, institucionalizados ou a viver em cidades [27]. Com base nestes casos, decidiu-se criar algumas personas e cenários específicos para descrever melhor a interação com o sistema de deteção de eventos significativos. Estes exemplos foram também importantes para a criação de uma plataforma para a geração automática de dados de sensores, servindo como base para as simulações efetuadas. Os dados a seguir apresentados foram revistos pelos gerontólogos envolvidos no projeto e aprovados pelos mesmos.

Persona 1: António, 73 anos, reformado, é viúvo e vive sozinho. Tem uma filha, a Maria, que vive a cerca de 1km e que visita o pai quase todos os dias ao final do dia, depois de sair do trabalho. Acorda de manhã, por volta das 8h, para tomar o pequeno-almoço. De seguida passa grande parte do dia fora de casa a cuidar dos animais e nas terras onde cultiva algumas frutas e legumes. Almoça em casa por volta das 13h, janta por volta das 20h e normalmente deita-se entre as 22h e as 23h. Aos domingos vai para a associação recreativa da parte da tarde, sendo que, por vezes só volta depois da hora de jantar.

O António sofre de apneia do sono, o que levanta algumas preocupações à sua filha. Para perceber se as rotinas de sono são afetadas, fazendo levantar-se mais cedo ou mais tarde que o normal, a filha instalou um sensor de pressão na cama para detetar quando é que ele se levanta. Além disso, com receio que esse problema afetasse a primeira refeição do dia, instalou um sensor de consumo na tomada do micro-ondas, onde aquece o leite todas as manhãs.

Num sábado, às 9:30h, a filha foi avisada que o António ainda não tinha acordado nem tomado o pequeno-almoço. Após uma visita a casa do pai percebeu que isto aconteceu porque ele tenha passado mal a noite e que por isso ficou a dormir até mais tarde.

Persona 2: Beatriz tem 80 anos e é viúva. Desde a morte do seu marido que vive com a filha, que tem 50 anos, o seu genro, que tem 52 anos e os seus dois netos com 22 e 19 anos. Acordam todos por volta das 7:30h para tomar o pequeno-almoço, porém, até às 9h já todos saíram de casa, ficando a Beatriz sozinha em casa até às 18h, exceto durante a hora de almoço (13-14h), altura em que a filha vai a casa. Desde que mudou de casa, apresentou sempre alguns sintomas de depressão como tristeza e falta de energia. Porém, nos últimos tempos, tem apresentado alguns sinais de demência, como o esquecimento da torneira da água a correr, algumas lembranças ou a troca do lugar das coisas.

Tendo em conta que a Beatriz não sai de casa enquanto todos estão fora, a ocorrência desse acontecimento deveria ser detetado rapidamente para perceber a razão de ter ocorrido. Para tal, foi instalado um sensor de contacto na porta principal e a um sensor de movimento para detetar se a porta se abria para a chegada de alguém ou para a saída da Beatriz.

Num dia da semana, por volta das 15 horas foi detetada uma abertura de porta anormal com o sensor de movimento a confirmar que alguém tinha saído de casa e deixado a porta aberta. A sua filha ao ver esta mensagem foi logo a casa ver o que se passava. Quase a chegar a casa, encontrou a mãe, com um ar perdido e assustado.

Persona 3: A Leonor, sexo feminino, tem 72 anos, é viúva e vive sozinha num apartamento em zona citadina. Tem um filho com 47 anos que vive no estrangeiro e vem visitá-la no Natal, na Páscoa, no aniversário e no verão. Acorda por volta das 8 horas e toma o pequeno-almoço fora de casa numa pastelaria perto de casa. Passa o resto da manhã fora de casa onde vai ao mercado municipal fazer as compras do dia ou ao cabeleireiro. Normalmente almoça depois das 13h e fica por casa a cuidar da casa, a ver televisão ou a tricotar. Janta por volta das 20h e deita-se por volta das 23h, no final da novela. Às quartas e sábados, das 21h às 23h tem ensaio do grupo de teatro e às sextas às 21h tem o ensaio do coro da igreja, que canta aos domingos de manhã às 11h.

Apesar de ter uma vida ativa e o filho ligar quase todos os dias, este está preocupado com algum imprevisto que possa acontecer de um momento para o outro. Como a sua mãe tem uma rotina bem definida, abrindo a janela do quarto, mesmo antes de tomar o pequeno-almoço, instalou um sensor de contacto na janela, para que fosse avisado caso essa rotina falhasse. A mesma coisa se aplicava ao fechar, porque permitia saber que estava em casa.

Por fim, instalou também um sensor de pressão na cama, que permitia saber quando é que ia dormir e se o fazia dentro da rotina, mesmo nos dias em que saísse à noite. Caso fosse detetado algo anormal, o filho seria informado e poderia ligar para saber o que se passava.

Um dia, a janela não abriu à hora habitual. O filho ao ser informado da situação e de ter estranhado decidiu ligar à mãe. Percebeu que isso aconteceu porque a Leonor tinha passado mal a noite, com tosse seca persistente e não conseguiu dormir em condições. Por isso, naquele dia tinha ficado na cama até um pouco mais tarde a descansar.

Persona 4: O Artur tem 83 anos, é viúvo, vive sozinho e tem uma filha de 56 anos. Acorda todos os dias por volta das 6:30h a carrinha da Santa Casa da Misericórdia passa por sua casa por volta das 10h para o levar para o centro de dia, onde almoça e passa o dia, e traz de volta às 17h. Depois janta por volta da 19h e às 21h já está a dormir. Aos fins-de-semana apanha um táxi para casa da filha, onde passa o dia e regressa ao início da noite já jantado. A filha visita o Artur regularmente ao fim do dia (18:30h) depois de sair do trabalho mas um dia encontrou os vizinhos do seu pai a ajudá-lo, porque estava alcoolizado e tinha tropeçado nos degraus da entrada da casa, tendo caído e não conseguindo-se levantar, devido ao seu peso e fraca mobilidade. Durante o período em que vai ao lar, ele pode sair para ir a vila conviver com alguns amigos de longa data e ingerir alguns copos de vinho. Isto ocorre com regularidade, o que deixa a filha preocupada com o que lhe pode acontecer enquanto está sozinho.

Para evitar a falta de resposta a esses problemas, instalou-se um sensor de pressão na cama, que permitia saber se o Artur ia dormir e se levantava à hora habitual. Caso o horário habitual não fosse cumprido, a filha era avisada e poderia passar por sua casa antes ou depois de ir trabalhar para saber se estava tudo bem.

Alguns meses após a instalação dos sensores, a filha do Artur recebeu uma mensagem às 22:30h, a avisar que não tinha sido detetada a presença do pai na cama e decidiu ver o que se passava. Quando chegou, encontrou o pai sentado no chão encostado ao sofá a dormir. Ele

estava alcoolizado e caiu na sala e após várias tentativas para se levantar, acabou por adormecer ali mesmo. A filha levou-o para a cama e resolveu uma situação que poderia ter ficado mais complicada se fosse detetada apenas no dia a seguir.

Persona 5: A Maria, 85 anos, reformada, é viúva e vive sozinha, numa zona rural. Tem dois filhos, mas vivem longe e apenas a visitam no fim-de-semana, altura em que levam as compras para a semana seguinte. Levanta-se por volta das 7:30h para ir buscar o pão que o padeiro deixou e volta para a cama de onde se levanta mais tarde, por volta das 10h, para tomar o pequeno-almoço e ver o programa da manhã. Por volta das 12h, o apoio domiciliário da Santa Casa da Misericórdia passa para deixar o almoço e jantar daquele dia e recolher os termos do dia anterior. Passa o dia todo em casa, deitada no sofá a ver televisão. Deita-se por volta das 0h, mas por vezes adormece no sofá antes de ir para a cama.

Para garantir que não há falhas na entrega do pão e assim a Maria possa tomar o pequeno-almoço, foram instalados sensores de movimento na parte exterior da porta, que detetavam a presença do padeiro, e sensores de contacto na porta, que detetavam o momento em que a Maria ia buscar o pão à porta.

Tendo em conta que a Maria está deitada toda a tarde e noite no sofá e que não tem mais nenhum tipo de apoio presencial até ao dia seguinte, será importante saber se acontece algo fora do normal, como uma queda, durante esse período. Para tal, foi instalado um sensor de pressão no sofá para detetar quando está deitada e avisar quando isso não ocorre durante um largo período de tempo. Também se instalou um sensor de pressão na cama, para o caso de ir para a cama no período em que não estiver no sofá e um sensor de consumo na televisão.

Um dia, foi enviada para a instituição uma mensagem que indicava que a Maria não estava na cama ou no sofá há mais de uma hora. Quando chegaram à sua habitação, perceberam que tinha tropeçado num degrau e caído perto da casa de banho e não tinha conseguido levantar-se porque lhe doía o joelho. Levaram-na ao hospital, onde pôde fazer todos os exames necessários e assim evitar que algo pior tivesse acontecido.

Persona 6: Julieta tem 77 anos e vive sozinha. Mora numa pequena aldeia rural no interior com cerca de 30 habitantes e tem 3 filhos, mas todos vivem no litoral e raramente a visitam. Ela levanta-se por volta das 8h para tomar o pequeno-almoço e de seguida vai tratar de alimentar os animais que tem e vai para a horta. Por volta das 12h passa o apoio domiciliário do município à população sénior que visita diariamente a aldeia para dar apoio com as contas, alimentação ou cuidados de saúde. Esta é a única monitorização que o município faz. Depois de almoço a Julieta dorme uma sesta e volta para a horta e alimenta os animais antes de anoitecer. Janta por volta das 19h, vê um pouco de televisão e deita-se por volta das 23h.

Porém um recente episódio em que Julieta saiu a meio da noite sem razão aparente deixou os vizinhos preocupados com um possível caso de demência. Para tentar evitar que isto voltasse a acontecer sem que fosse detetado, instalaram-se sensores de contacto nas portas de acesso ao exterior e de movimento na parte exterior da casa, que iam permitir saber se alguém teria saído a meio da noite da casa.

Por fim, a instituição de apoio queria garantir que os seus utentes recebiam apoio diário nestas visitas. Para isso, ofereceu um porta-chaves com uma *itag bluetooth* para detetar quando um funcionário ia a casa de um utente e quando falhava essa visita.

Um dia, esse apoio falhou e foi enviada uma mensagem para a instituição por volta das 13h. Para colmatar essa falha, a visita foi realizada da parte da tarde.

Uns dias mais tarde, durante a noite, os sensores de porta e de movimento detetaram atividade anormal às 4h. A equipa de apoio da instituição, após ser informada, interveio de imediato. Quando chegaram, encontraram a Beatriz perto de uma árvore a apanhar fruta. Mostrava sinais de esquecimento e de hipotermia. A rápida intervenção permitiu levá-la ao hospital para fazer exames e evitar males maiores.

Nos pontos subsequentes pretende-se que, a partir dos casos anteriormente enumerados, seja possível definir os casos de uso do sistema, os atributos de qualidade e as restrições associadas ao projeto.

3.2 Requisitos funcionais

O requisito é uma condição necessária para a consecução de um fim. Neste projeto, pretende-se reunir um conjunto de especificações que sejam capazes de dar resposta ao desejado para o produto final e que cumpra com os objetivos inicialmente definidos. Na ***Componente de deteção de eventos significativos*** irá decorrer todo o processo de tratamento e análise dos dados dos sensores instalados nas casas dos utentes. Os atores e os casos de uso serão descritos em detalhe nas subsecções seguintes.

3.2.1 Atores

Os atores representam os intervenientes externos ao sistema que vão interagir com o ambiente que será desenvolvido. Para este diagrama de casos de uso foram definidos dois tipos diferentes de atores:

- **Gateway IoT** - Este ator será responsável por recolher os dados dos sensores instalados nas casas dos idosos e, posteriormente, enviá-los para a Componente de deteção de eventos significativos para serem processados.
- **Relógio** – O Relógio será responsável por, periodicamente, verificar que existem alarmes que serão posteriormente notificados ao Sistema de notificações
- **Sistema de notificações** - Com base no tratamento dos dados dos sensores recebidos, o Sistema de notificações será o ator responsável por comunicar aos cuidadores os eventos significativos detetados.

3.2.2 Casos de uso

Um caso de uso refere-se a uma ação que vai ser executada no ambiente, por um determinado ator e que vai ter consequências em todo o sistema. Na tabela apresentada de seguida são descritos os casos de uso necessários implementar no sistema, de forma a responder aos problemas inicialmente identificados. Com base no método *standard* MoSCoW foi atribuída uma prioridade por cada caso de uso descrito, de forma a estabelecer aqueles que deverão ser desenvolvidos nas primeiras iterações. Essa classificação poderá variar entre:

- ***Must have*** - Fazem parte os casos de uso principais e os mais comuns no sistema.
- ***Should have*** - Fazem parte os casos de uso menos importantes para o sistema.
- ***Could have*** - São os casos de uso que são desejáveis, mas não necessários.
- ***Won't have*** – Correspondem aos casos de uso que ficam para uma entrega futura.

Na Figura 3.1 é apresentado o diagrama de casos de uso da componente deteção de eventos significativos do SOCIAL, que descreve de forma sucinta a funcionalidade proposta. Os casos de uso são diferenciados pelas prioridades *Must have*, *Should have* e *Could have e Won't have* no projeto, às quais correspondem, respetivamente, as cores verde, amarela, laranja e vermelha.

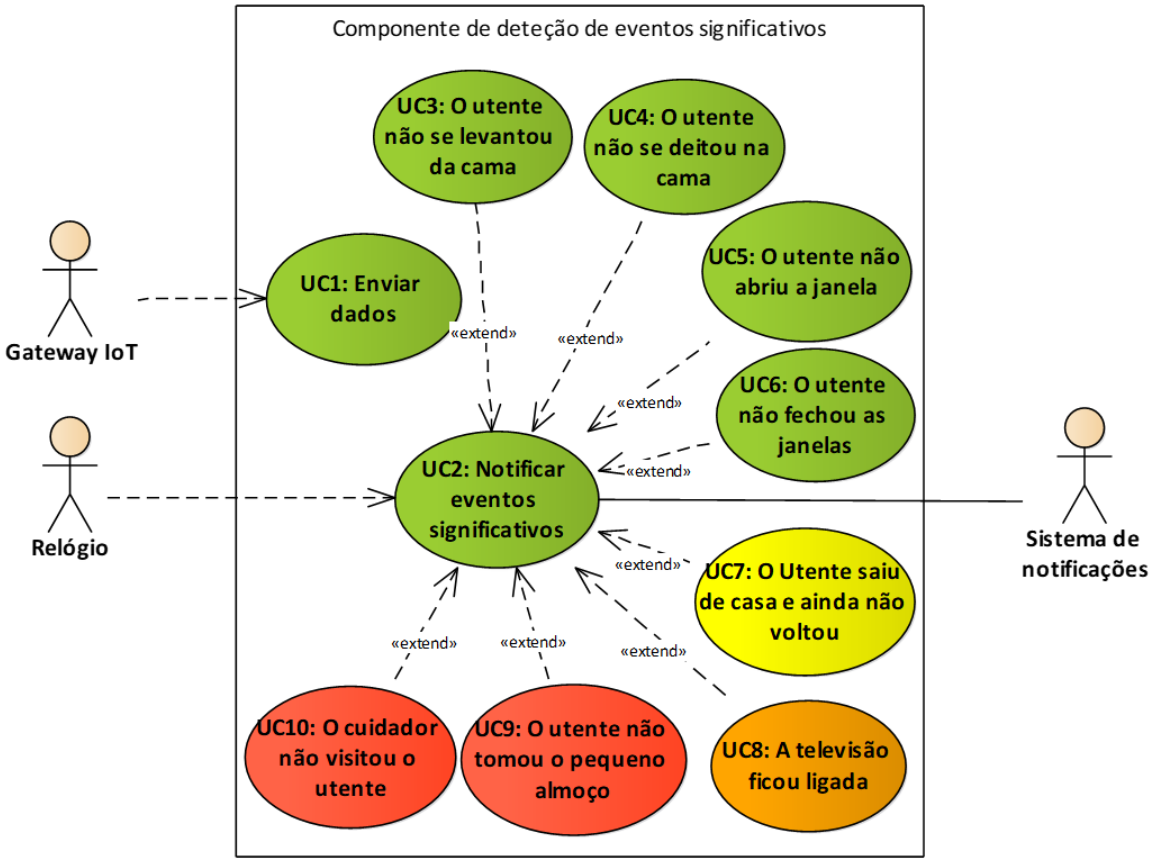

Figura 3.1 - Diagrama de casos de uso.

Tabela 3.1 - Descrição dos casos de uso identificados.

ID – Título	Descrição	Prioridade
UC1 - Enviar dados	Este caso de uso é essencial para que todo o componente a desenvolver seja capaz de funcionar. A Gateway IoT irá tratar de enviar os dados dos sensores para que o sistema os possa processar e determinar os eventos significativos.	*Must have*

ID – Título	Descrição	Prioridade
UC2 - Notificar eventos significativos	Após o Relógio verificar a existência de alarmes, será necessário comunicá-los aos cuidadores. Esta tarefa será executada pelo Sistema de notificações e poderá avisar o cuidador sobre qualquer uma das situações enumeradas nos casos de uso seguintes, caso estes se verifiquem.	*Must have*
UC3 - O utente não se levantou da cama	Este caso de uso ajuda a determinar se o utente ainda não acordou. Caso se verifique que ainda não se levantou dentro de uma hora considerada "normal" para o utente em questão, deverá ser lançada uma notificação ao cuidador a relatar a situação. Para a deteção desta situação deverá ser utilizado um sensor de pressão na cama e conjugar com o horário atual.	*Must have*
UC4 - O utente não se deitou na cama	Este caso de uso é semelhante ao descrito anteriormente (UC3), porém, refere-se ao facto de um utente não se deitar na sua cama à noite dentro do horário habitual. Deverá ser notificado o cuidador e deverá ser utilizado também um sensor de pressão na cama.	*Must have*
UC5 - O utente não abriu a janela	Este caso de uso é importante para detetar quando o utente acorda. Para tal, será necessário instalar no seu quarto um sensor de contacto na janela, que vai permitir saber quando é que está aberta ou fechada. Desta forma será possível detetar anomalias na rotina do utente e intervir, caso exista necessidade.	*Must have*
UC6 - O utente não fechou as janelas	Este caso de uso é específico para quando um utente deixa as janelas abertas durante a noite. Como é algo que não é normal, deverá ser lançada uma notificação a avisar o cuidador. Para a deteção deste caso de uso será necessária a instalação de sensores de contacto na janela, de modo a determinar o seu fecho ou abertura e conjugar com o horário em que a janela está aberta.	*Must have*

ID – Título	Descrição	Prioridade
UC7 - O utente saiu de casa e ainda não voltou	Este caso é especialmente importante para pessoas que apresentem problemas de memória e que não consigam regressar a casa. Deverão ser utilizados os sensores de movimento para detetar a presença do utente em casa e sensores de contacto nas portas para identificar a saída e o retorno a casa do mesmo. Caso o utente não regresse a casa, deverá ser enviada uma notificação para o cuidador.	*Should have*
UC8 - A televisão ficou ligada	Este caso de uso é importante para determinar se o utente foi dormir e deixou a televisão ligada. Deverá ser utilizado um sensor de consumo para a televisão, para que seja capaz de comparar com o histórico de utilização e enviar alertas caso se verifique este evento.	*Could have*
UC9 - O utente não tomou o pequeno-almoço	Este caso de uso permite saber se o utente tomou o pequeno-almoço. É necessária a instalação de sensores de consumo nas tomadas dos equipamentos utilizados habitualmente para o pequeno-almoço (micro-ondas, torradeira). Caso não seja detetado qualquer tipo de atividade por parte destes equipamentos será lançada uma notificação para o cuidador a avisar da situação.	*Won't have*
UC10 - O cuidador não visitou o utente	Este caso de uso é específico para utentes que habitualmente não abandonam a sua casa. Através da instalação de um sensor de contacto na porta de entrada principal da casa e atribuindo um sensor (iTag) ao cuidador é possível detetar quando um cuidador visita o utente e encontrar anomalias, no caso de existir uma visita regular por parte do cuidador.	*Won't have*

3.3 Atributos de Qualidade

Os atributos de qualidade são propriedades do *software* que se podem manifestar como capacidades, características ou restrições de uma função específica ou de um conjunto de funções [28] e estão diretamente ligados a propriedades que vão afetar a utilização do

sistema. Assim, tanto podem interessar ao utilizador, no caso de atributos relacionados com a usabilidade e o desempenho, bem como à equipa de desenvolvimento, para atributos relacionados com custos e planeamento.

Para este projeto, foram definidos atributos de qualidade com base nos casos de uso descritos anteriormente. Na Tabela 3.2 é possível observar cada um deles, bem como a sua descrição.

Tabela 3.2 - Descrição dos atributos de qualidade do projeto.

ID - Atributo de qualidade	Descrição
AQ1 – O Velocidade de processamento dos dados recebidos dos sensores	As *gateways* instaladas nas casas de 100 utentes enviam a cada minuto os valores medidos pelos sensores associados a cada uma (4 sensores por casa em média com três séries de dados por sensor). O *Midlleware* de IoT e o sistema de deteção de eventos significativos estão a operar normalmente. Em média o sistema não demora mais do que 50 milissegundos a processar cada série de dados.
AQ2 - Envio de notificações para o cuidador no menor tempo possível	O sistema deteta um desvio ao padrão de deitar e levantar do utilizador. Os sistemas de deteção de eventos significativos e de notificações estão a operar normalmente. É enviado para o Sistema de notificações um alerta que fará o envio de uma mensagem para o cuidador no tempo máximo de 10 minutos, para 80% dos casos.
AQ3 - Baixo número de falsos negativos e falsos positivos	O sistema deteta um desvio ao padrão de deitar e levantar do utilizador. Os sistemas de deteção de eventos significativos e de notificações estão a operar normalmente. De forma a reduzir o número de falsos positivos e falsos negativos, é dado algum intervalo de tempo para o utente levantar ou deitar. É enviado para o Sistema de notificações um alerta que fará o envio de uma mensagem para o cuidador no tempo máximo de 60 minutos, para 80% dos casos.

ID - Atributo de qualidade	Descrição
AQ4 - Extensibilidade	Ao adicionar uma série de dados nova não é necessário alterar a interface do sistema. Porém é necessário construir um novo filtro, com novas funcionalidades, que deverá ser implementado pelo programador. Esta alteração no sistema deverá ser implementada num prazo máximo de 40 horas em 75% dos casos.
AQ5 - Modificabilidade	A adição de um algoritmo novo ao sistema leva a uma modificação no sistema, apesar de não ser necessário alterar a interface. A nova funcionalidade deverá ser implementada por um programador e não deverá demorar mais do que 80 horas em 80% dos casos para concluir todo o processo

3.4 Restrições

As restrições são limitações impostas ao projeto e correspondem a decisões que normalmente são tomadas numa parte inicial do projeto e que podem ter origem no cliente, na entidade que o irá desenvolver ou no ambiente de aplicação.

Na Tabela 3.3 apresentam-se as restrições de negócio do projeto, que não tem implicação direta na arquitetura, mas definem como serão executados os processos do sistema. Habitualmente estão associados a prazos e a recursos.

Tabela 3.3 - Descrição das restrições de negócio do projeto.

ID - Restrição	Descrição
RN1 - Data limite de desenvolvimento	A Componente de deteção de eventos significativos deverá estar concluída até ao dia 30 de Julho.
RN2 - Recursos humanos do projeto	O projeto requer uma pessoa a trabalhar 20 horas semanais, entre 18 de Setembro e 2 de Fevereiro, e 40 horas semanais entre 5 de Fevereiro e 30 de Junho.

Na Tabela 3.4 são apresentadas as restrições técnicas do projeto. Estas limitações já apresentam um impacto direto na arquitetura e definem o *software* e *hardware* que pretende ser utilizado no sistema.

Tabela 3.4 - Descrição das restrições técnicas do projeto.

ID - Restrição	Descrição
RT1 - Utilizar a linguagem Java	Numa fase inicial do projeto, o consórcio decidiu que as componentes a desenvolver deveriam ser implementadas na linguagem Java.
RT2 - Utilizar protocolo de mensagens de implementação assíncrona	O protocolo escolhido deverá utilizar implementação assíncrona, que é mais indicado para dispositivos de IoT, pois permite uma maior escalabilidade.
RT3 - Utilizar protocolo de formato mensagem *standard*	O protocolo escolhido terá que ser padronizado, de forma a permitir que o formato das mensagens esteja previamente definido e permita às várias componentes comunicarem entre si.
RT4 - Persistência de dados em recursos FHIR	Definido previamente pelo líder do consórcio, a persistência de eventos significativos deverá ser feita utilizando recursos FHIR.
RT5 - Utilização do protocolo XACML para a autorização	Esta componente é desenvolvida por outro parceiro do consórcio, que decidiu adotar estes protocolos para a autorização dos dados da Componente de deteção de eventos significativos.

Capítulo 4
Estrutura do projeto e experiências preliminares

4.1 Visão geral

A plataforma SOCIAL não é feita apenas da Componente de deteção de eventos significativos e apresenta um vasto conjunto de elementos que necessitam ser desenvolvidos. A Figura 4.1 apresenta duas camadas superiores da arquitetura:

- **Camada de aplicação** - É a camada mais próxima das aplicações do utilizador, como os dispositivos IoT e as aplicações Mobile e *Web*.
- **Camada de negócio** - Também conhecida por camada lógica, vai ser composta pelas componentes que definem as regras do sistema, como é o caso das componentes de tratamento dos dados dos sensores, da autenticação e da autorização.

O âmbito do trabalho a desenvolver no projeto está bem definido e corresponde à componente para deteção de eventos significativos. Presente na camada de negócio, está identificada na figura seguinte com um círculo vermelho como **Significant Event Detection**.

Na Figura 4.1 é possível observar que os dispositivos IoT (sensores) enviam os dados para uma máquina intermédia (*Gateway Social*), que fará, posteriormente, o envio para a Componente de deteção de eventos significativos. Após o tratamento dos dados, será necessário autorizar os dados para que estes possam ser armazenados como recursos FHIR (FHIR Gateway), caso sejam detetados eventos significativos. Desta forma, este trabalho fica confinado ao processamento de dados e identificação de padrões. Após essa tarefa, os eventos deverão ser lançados para gerar notificações e serem guardados, mas essas tarefas já não fazem parte do âmbito do trabalho de estágio.

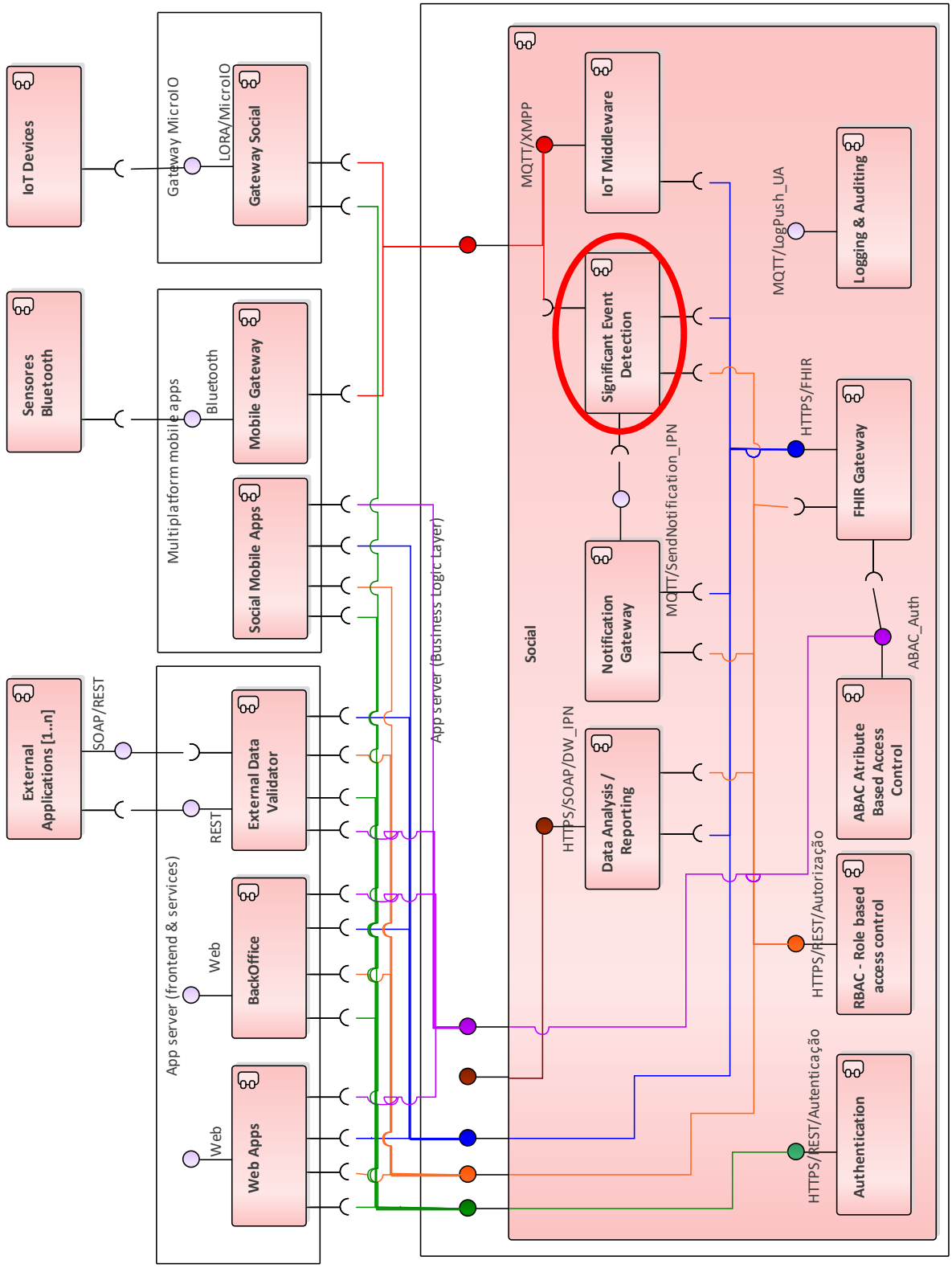

Figura 4.1 - Localização da Componente de deteção de eventos significativos na arquitetura da plataforma SOCIAL.

4.2 Descrição da Componente de deteção de eventos significativos

Descritas as interações dos outros componentes da plataforma SOCIAL com aquele que se pretende desenvolver, é necessário perceber como é que a Componente de deteção de eventos significativos irá funcionar.

O sistema contará com um elemento que estará à escuta de dados dos sensores enviados pelo *Gateway,* designado como "*Receiver*". Este irá armazenar grandes quantidades de dados por períodos de tempo que podem ser definidos pelo programador, o que irá dar origem a um conjunto de dados chamado RDD (*Resilient Distributed Datasets*) [29]. Eles são a estrutura de dados fundamental do *Spark* e permitem que cada conjunto seja dividido em diferentes partições. Desta forma, será possível fazer o tratamento dos dados em paralelo, tornando o processo mais rápido.

Para cumprir com a tarefa de detetar eventos significativos é necessário recorrer a uma base de dados com o histórico de um utente. Desta forma, conjugado com os dados recebidos, será possível fazer essa análise e o cuidador ser notificado em caso de necessidade. Concluída a análise, os novos elementos são enviados para a base de dados para serem utilizados em futuras pesquisas.

4.3 Seleção das tecnologias

Nesta secção vão ser descritas as tecnologias necessárias para a implementação desta componente. Das restrições descritas no Capítulo 3 é possível identificar duas tecnologias de uso obrigatório:

- **Java** - Linguagem de programação orientada a objetos que servirá de base para a implementação de todo o código desenvolvido.
- **FHIR** - O FHIR (*Fast Healthcare Interoperability Resources*) é uma especificação de plataforma, que define um conjunto de recursos direcionados para o processo de assistência médica [30]. No projeto SOCIAL pretende-se utilizar os recursos que normalmente são usados no âmbito dos cuidados formais também para os cuidados informais. Esta especificação padrão será necessária para a persistência de eventos significativos detetados pela componente desenvolvida no âmbito do estágio.

4.3.1 Ferramenta para processamento de dados em tempo real

Relativamente à ferramenta que vai tratar os dados recebidos dos sensores em tempo real, a escolha recaiu sobre o ***Spark Streaming***. Esta tecnologia, associada ao ***Apache Spark***, permite o processamento da grande quantidade de dados que vai ser recebida e num curto espaço de tempo, inferior ao da plataforma *Hadoop* [31]. Para além da compatibilidade com a linguagem Java, o *Spark Streaming* apresenta também as seguintes vantagens [32]:

- Recuperação rápida de falhas.
- Bom balanceamento de carga e uso de recursos.
- Combinação de dados em tempo real com conjuntos de dados estáticos e consultas interativas.
- Integração nativa com bibliotecas de processamento avançado (SQL, aprendizagem computacional, processamento de grafos).

Estas são características que se enquadram naquilo que é pretendido para este projeto e que fizeram com que esta ferramenta fosse a escolhida, ao invés de outras analisadas, como ***Flink, Kafka Streams*** ou ***Beam***. No geral, essas plataformas eram um pouco mais complexas e não tinham tanta informação disponível, comparativamente com o *Spark Streaming*.

4.3.2 Protocolo de comunicação

Um dos requisitos enumerados anteriormente fazia referência à escolha de um protocolo de comunicação assíncrono, ou seja, os dados podem ser transmitidos com qualquer ordem. Para cumprir esta condição foi escolhido o protocolo **MQTT** (*Message Queuing Telemetry Transport*), que se ajusta às condições do projeto, porque o transporte das mensagens é bastante leve, é usado para comunicações com locais remotos e é direcionado para sensores e pequenos dispositivos móveis [33].

4.3.3 Protocolo de mensagens

Quanto ao formato das mensagens enviadas entre a componente *Gateway* e a componente do projeto será utilizado o **XMPP** (*Extensible Messaging and Presence Protocol*). Para além de ser um protocolo *standard*, respeitando uma das restrições do projeto, permite também definir o formato da mensagem, uma das vantagens comparativamente ao MQTT [34]. A estrutura definida ajudará a validar as mensagens recebidas e será mais fácil tratar e perceber os dados

enviados pelos sensores. Outra das vantagens relativamente ao MQTT é que é mais escalável à medida que é aumentado o número de dispositivos implementados.

4.3.4 Base de dados

Para este projeto é necessária uma base de dados capaz de lidar com o problema de *Big Data*, que é a gestão do armazenamento de grandes quantidades de informação. Essa foi uma das principais razões para se optar pela escolha de uma base de dados **NoSQL.** Esta opção não usa SQL, não seguindo o modelo relacional, tornando este tipo de sistema mais flexível e, assim, facilitando a adição de dados sem reestruturar todas as tabelas [35]. A decisão do tipo de base de dados a utilizar foi acordada entre todos os elementos do consórcio.

4.4 Persistência dos dados em FHIR

Apesar de ser um ponto que não faz parte do trabalho a realizar no âmbito do estágio, é importante perceber de que forma os dados dos sensores e os eventos significativos vão ser persistidos em FHIR e as diferenças entre os diferentes recursos que se irão utilizar. São os seguintes:

- ***Device*** – Este recurso será utilizado para persistir os sensores e as suas informações. Os *Devices* criados vão corresponder aos sensores que serão instalados nas casas dos pacientes. Para além disso serão persistidos sensores virtuais, que correspondem aos filtros de deteção de eventos. Os campos relevantes utilizados no recurso *Device* são:
    - ***identifier*** – Contém o IMEI (*International Mobile Equipment Identity*) do sensor para ser identificado no sistema;
    - ***type*** - Tipo de sensor representado, onde no caso dos sensores virtuais discutidos anteriormente será utilizado o tipo "*Experimental Device*";
    - ***status*** - Representa se o dispositivo está ativo ou não;
    - ***model*** **-** Será utilizado para representar a classe do filtro para saber que leitura este sensor virtual fornece;
    - ***note*** - Utilizado para inserir a configuração do filtro;
    - ***patient***: Utente associado ao sensor.
- ***Observation*** – Este recurso será utilizado para gravar os eventos detetados pelo Sistema de Deteção de Eventos. Os campos relevantes a serem utilizados são:
    - ***issued*** **-** Hora em que o evento aconteceu;

- ***value*** - Resultado do evento;
- ***interpretation*** - Utilizado para validar a informação sobre o evento notificado aos cuidadores. São utilizados os valores "*Detected*", "*Positive*" e "*Negative*";
- ***subject*** - Utente sobre o qual o evento foi detetado;
- ***device*** - Sensor ou sensor virtual que detetou o evento;
- ***code*** - Tipo de evento que foi detetado.

- ***CommunicationRequest*** – O recurso Communication Request será utilizado para gravar os alertas gerados pela deteção de um evento significativo. Os campos relevantes a serem utilizados são:
  - ***recipient*** - representa a equipa de cuidados do utente;
  - ***subject*** - utente associado ao alerta;
  - ***reasonReference*** - referência ao evento (recurso *Observation*) que foi persistido no FHIR.

Os cuidadores serão notificados através da aplicação "*My SOCIAL",* onde poderão fazer uma avaliação sobre o alerta de evento significativo que receberam, de forma a perceber se corresponde à realidade. Os valores "*Detected*" e "*Positive*" do campo *interpretation* serão tratados como eventos detetados corretamente e "*Negative*" como falsos positivos. Esse julgamento permitirá mais tarde treinar o algoritmo de deteção de eventos por aprendizagem com base nos dados que foram recolhidos e avaliados e, desta forma, baixar o número de falsos positivos e falsos negativos.

4.5 Segurança e privacidade

Para garantir a privacidade dos dados foi usado o modelo *standard* XACML (*eXtensible Access Control Markup Language*) [36]. Esta especificação é composta por dois grandes componentes: o ABAC (*Attribute-Based Access Control*) que é um sistema de controlo dos dados baseado em atributos e o RBAC (*Role-Based Access Control*) que é um sistema de controlo de dados baseado na função do utilizador. Desta forma, é possível criar regras bem definidas que possibilitem aos diferentes atores do sistema acederem a determinados dados, tendo em conta a sua função e os utentes que se relacionam com esse utilizador.

Neste processo, o utilizador envia um pedido para o PEP (*Policy Enforcement Point*) para que se possa iniciar o processo de validação para o acesso aos dados. Nesse pedido é necessário enviar informação sobre a pessoa que está a tentar aceder a um determinado conteúdo, o

recurso FHIR à qual tenta o acesso, a ação ou ações que o utilizador pode realizar e o ambiente, que pode representar diversos aspetos do pedido, diferindo assim de pedido para pedido. Pode conter informação acerca da organização que representa, um conjunto de campos associados aos atributos de um recurso, um campo em específico para saber se pode visualizar ou não esse campo, entre outros.

Esta componente de segurança não faz parte do âmbito do trabalho a realizar e o seu desenvolvimento está a cargo da Universidade de Aveiro.

4.6 Experiências preliminares

Neste projeto, a realização de experiências apresenta um papel fundamental naquilo que será o desenvolvimento da arquitetura e do projeto em geral. Para a concretização com sucesso dos objetivos desta componente, é necessário conhecer bem as ferramentas que vão ser utilizadas e perceber se são as mais indicadas para o fim que se pretende.

No final destas experiências pretendia-se que estivessem reunidos os conhecimentos básicos para avançar no desenvolvimento do projeto e que a última experiência fosse o mais semelhante possível à Componente de deteção de eventos significativos, quer no formato de dados a utilizar, quer na forma de tratar os mesmos com o *Spark Streaming*.

Assim sendo, foram planeadas quatro experiências relacionadas com o tratamento e análise de dados, divididas da seguinte forma:

- **Experiência 1 - Experiência com *Spark/Spark Streming*** - O principal objetivo desta experiência era a aquisição de conhecimentos sobre esta tecnologia, a partir da realização de um tutorial. No final pretendia-se que um programa emissor fosse capaz de ler dados de um ficheiro de texto e os enviasse para o programa recetor.
- **Experiência 2 - Experiência com *Message Queues*** - O objetivo desta experiência passava por perceber se era possível reproduzir o mesmo programa efetuado na experiência 1, mas introduzindo a utilização de *message queues* para efetuar a comunicação entre emissor e recetor.
- **Experiência 3 - Experiência com um sensor de presença (*iTag*)** - Para esta experiência pretendia-se perceber se, a partir daquilo que tinha sido executado na experiência 2, era possível fazer a leitura periódica dos dados de sensores, guardados num ficheiro de texto, e se o recetor era capaz de fazer algum tipo de processamento dos mesmos.

- **Experiência 4 - Experiência com a utilização de dados no formato *standard* (XMPP)** - Para esta experiência está planeada a utilização de dados reais de alguns sensores, para que o formato XMPP possa ser validado.

Os resultados das experiências realizadas foram de enorme importância para o projeto, porque permitiram mitigar uma série de riscos que poderiam ter um impacto catastrófico, caso não fossem detetados nesta fase. Os problemas com incompatibilidades tecnológicas no projeto não vão existir, porque foi provado que é possível construir um sistema equivalente ao protótipo que se pretende desenvolver. Ao realizar as experiências, foi também reduzido o risco de não ser possível avançar por causa da falta de conhecimento de alguma tecnologia ou protocolo. Além disso, a comprovada utilização de protocolos *standard*, mais especificamente para o formato das mensagens, permite a introdução de vários tipos de sensores para além daqueles especificamente definidos para este projeto.

No âmbito do estágio curricular ficou definida a implementação desta componente que, a partir dos dados que vão ser gerados a partir de um simulador, seja capaz de identificar eventos e detetar padrões. Porém, a utilização do *Spark* será importante no processo de implementação futura no projeto SOCIAL, sendo essencial a realização deste trabalho experimental.

Com base no modelo para elaboração de experiências, presente em [37], são apresentadas as quatro experiências realizadas de forma mais detalhada.

Tabela 4.1 - Experiência com Spark /Spark Streaming.

Pré-experiência	
Título: Experiência com *Spark* /*Spark Streaming*	**ID:** 1
ID do problema/Descrição: Falta de conhecimentos sobre as ferramentas *Spark* e *Spark Streaming*.	**Responsável**: João Rainho
Razão do problema (escolher uma): ☐ Reparar, atualizar e/ou esclarecer a documentação (texto, esquemas, tabelas e assim por diante). ☐Necessária mais informação técnica. ☐Necessária mais informação sobre elementos da arquitetura. ☒Experiência necessária. ☐Outros (descrever):_____	
Objetivo: O objetivo desta experiência passa por aprender a linguagem *Spark*/*Spark Streaming*, que será utilizada no âmbito do projeto para o tratamento e análise dos dados recebidos pelos sensores.	
Descrição da experiência: A experiência consistirá na realização de um tutorial online de *Spark* e *Spark Streaming*, permitindo no final do mesmo ter um programa emissor capaz de ler dados de um ficheiro de texto e a comunicar através do protocolo TCP com um programa recetor que vai tratar os dados.	
Artefactos criados: Código para comunicação entre emissor e recetor, bem como código para análise e tratamento dos dados recebidos do emissor.	
Critérios de conclusão: Os dois programas comunicam entre si pelo canal TCP e os dados são tratados com sucesso pelo recetor com *Spark Streaming*.	
Recursos necessários: Um computador capaz de executar programas em Java e ficheiro de dados para teste.	
Duração estimada e *milestones* principais: 40 horas	
Pós experiência	
Resumo das descobertas: O *Spark Streaming* permite agrupar os dados recebidos pelo emissor durante um período de tempo para os tratar posteriormente em conjunto. Além disso, o *Spark Steaming* permite filtrar a informação recebida.	
Duração atual: 34 horas	
Recursos atuais: Os mesmos, antes de iniciar a experiência.	
Recomendações do responsável: O agrupamento de dados do *Spark*/*Spark Streaming* vai facilitar a análise futura dos dados dos sensores, permitindo detetar um maior número de falsos positivos. Com a opção de filtrar, também será possível selecionar o tipo de dados que vamos querer analisar.	

Tabela 4.2 - Experiência com *Message Queues*.

Pré-experiência	
Título: Experiência com *Message Queues*	**ID:** 2
ID do problema/Descrição: Necessidade de utilizar as *Message Queues* no âmbito do projeto.	**Responsável: João Rainho**
Razão do problema (escolher uma): ☐ Reparar, atualizar e/ou esclarecer a documentação (texto, esquemas, tabelas e assim por diante). ☐Necessária mais informação técnica. ☐Necessária mais informação sobre elementos da arquitetura. ☒Experiência necessária. ☐Outros (descrever):_____	
Objetivo: O objetivo desta experiência passa por conseguir transmitir os dados entre emissor e recetor a partir de uma *Message Queue*.	
Descrição da experiência: A experiência consistirá em aproveitar o código construído na experiência 1 e substituir a comunicação TCP pelas *Message Queues* que serão utilizadas no projeto.	
Artefactos criados: Código para o envio dos dados através de uma *Message Queue*.	
Critérios de conclusão: Os dois programas comunicam entre si através da *Message Queue* e os dados são tratados com sucesso pelo recetor com *Spark Streaming*.	
Recursos necessários: Um computador capaz de executar programas em Java e ficheiro de dados para teste.	
Duração estimada e *milestones* principais: 20 horas	
Pós experiência	
Resumo das descobertas: A utilização de *message queues* não compromete o funcionamento desejado do programa desenvolvido.	
Duração atual: 20 horas	
Recursos atuais: Os mesmos antes de iniciar a experiência.	
Recomendações do responsável: Manter a utilização das *message queue* nas experiências seguintes.	

Tabela 4.3 - Experiência com um sensor de presença (iTag).

Pré-experiência	
Título: Experiência com um sensor de presença (*iTag*)	**ID:** 3
ID do problema/Descrição: É necessário compreender se este modelo é capaz de tratar dados recebidos a partir de sensores.	**Responsável**: João Rainho
Razão do problema (escolher uma): ☐ Reparar, atualizar e/ou esclarecer a documentação (texto, esquemas, tabelas e assim por diante). ☐Necessária mais informação técnica. ☐Necessária mais informação sobre elementos da arquitetura. ☒Experiência necessária. ☐Outros (descrever):_____	
Objetivo: Os objetivos desta experiência passam por simular a utilização de sensores e adaptar o código do programa recetor à leitura dos dados enviados pelo programa emissor.	
Descrição da experiência: A experiência consistirá em aproveitar o código construído na experiência 2 e adaptá-lo às características da informação enviada pelo sensor. Para tal, o programa emissor vai ler de um ficheiro de texto os dados sobre o sensor de presença, com uma periodicidade de 2 segundos, e enviará para o recetor, que irá analisar os dados e reconhecer se o sensor naquele momento está visível ou não.	
Artefactos criados: Código do recetor capaz de tratar os novos tipos de dados que vai receber.	
Critérios de conclusão: Os programas são capazes de comunicar entre si e o recetor é capaz de perceber se o sensor de presença está visível ou não.	
Recursos necessários: Um computador capaz de executar programas em Java e ficheiro de texto com dados de sensores para teste.	
Duração estimada e *milestones* principais: 20 horas	
Pós experiência	
Resumo das descobertas: O tratamento dos dados foi feito com sucesso, porém será necessário adaptar a outros sensores e com os dados num formato *standard*.	
Duração atual: 24 horas	
Recursos atuais: Os mesmos antes de iniciar a experiência.	
Recomendações do responsável: É recomendada a utilização de um formato de mensagens *standard*, desta forma, o tratamento de dados será uniforme e mais simpres de realizar.	

Tabela 4.4 - Utilização de dados no formato standard (XMPP).

Pré-experiência	
Título: Utilização de dados no formato standard (XMPP)	**ID:** 4
ID do problema/Descrição: Para o projeto, os dados dos sensores vão estar no formato standard XMPP.	**Responsável: João Rainho**
Razão do problema (escolher uma): ☐ Reparar, atualizar e/ou esclarecer a documentação (texto, esquemas, tabelas e assim por diante). ☐Necessária mais informação técnica. ☐Necessária mais informação sobre elementos da arquitetura. ☒Experiência necessária. ☐Outros (descrever):_____	
Objetivo: Perceber o funcionamento do formato *standard* XMPP e adaptar.	
Descrição da experiência: A experiência consistirá em aproveitar o código construído na experiência 3 e substituir o formato dos dados que vão ser recolhidos para o formato XMPP, utilizado no âmbito do projeto.	
Artefactos criados: Código do recetor capaz de tratar os novos tipos de dados que vai receber.	
Critérios de conclusão: Os programas são capazes de comunicar entre si e o recetor é capaz de perceber se o sensor de presença está visível ou não, utilizando este formato de dados.	
Recursos necessários: Um computador capaz de executar programas em Java e ficheiro de texto com dados de sensores no formato XMPP para teste.	
Duração estimada e *milestones* principais: 20 horas	
Pós experiência	
Resumo das descobertas: O tratamento dos dados foi feito com sucesso e o protocolo pode ser utilizado na implementação futura	
Duração atual: 28 horas	
Recursos atuais: Os mesmos antes de iniciar a experiência.	
Recomendações do responsável: Nenhuma	

Capítulo 5
Arquitetura

A definição da arquitetura de um sistema de *software* é um processo complexo que exige uma grande responsabilidade. A documentação e diagramas criados permitem fazer a ligação dos requisitos à implementação que será executada numa fase posterior. Por essa razão é importante que seja criada uma arquitetura simples, mas completa, para compreender mais facilmente o projeto e as componentes que o integram. Caso contrário, se a arquitetura apresentar ambiguidades, poderão surgir problemas e condicionar o resultado final.

Para este projeto será utilizado o modelo ***Risk Driven Architecture***. Neste modelo pretende-se aplicar um conjunto de técnicas arquiteturais, de modo a reduzir os riscos, que se baseiam em três etapas [38]:

- **Identificar e priorizar os riscos** - Perceber quais as ameaças associadas ao projeto e classificar as que poderão ter um maior impacto.
- **Selecionar e aplicar um conjunto de técnicas** - Encontrar as técnicas que poderão evitar a concretização dos riscos.
- **Avaliar a evolução dos riscos** - Analisar de forma regular os riscos e alterar as suas prioridades, caso seja necessário.

Avançar no projeto sem perceber a arquitetura do sistema, sem ter a certeza que as tecnologias escolhidas são as melhores ou com a possibilidade de não ser possível a sua utilização acabaria sempre por ter um impacto catastrófico no resultado final. Para evitar que tal ocorra, foram traçados os diagramas arquiteturais do sistema e comparadas as tecnologias que mais se adequavam. Também foram desenhadas experiências que, para além de servirem como forma de aprendizagem nos pontos onde havia lacunas de conhecimento, permitiam demostrar que era possível utilizá-las no contexto do projeto.

5.1 Arquitetura da componente deteção de eventos significativos

O sistema de deteção de eventos significativos segue uma arquitetura *Pipe and Filter*. Esta componente é alimentada por uma fonte de dados, que no caso deste projeto são as *gateways*. Para tratar os dados recebidos, este sistema vai recorrer a um conjunto de filtros que serão capazes de trabalhar em paralelo, facilitando o processamento dos dados [39]. Os dados obtidos vão depender do tipo de evento que se pretende detetar e, consequentemente, dos filtros usados. Posteriormente, esses dados serão enviados para outras componentes, que os vão usar para enviar notificações e guardar na base de dados. Outra das vantagens desta arquitetura, para além da execução concorrente, é a simplicidade de implementação e a capacidade de reutilizar os filtros em futuras iterações.

Este sistema pode ser dividido em três blocos, como é demonstrado na Figura 5.1. O primeiro corresponde à fonte de dados (*Source*), que emite os dados do sensor e alimenta o bloco *Filter*, através de um canal de comunicação para a transmissão de informação. O segundo bloco contém todos os filtros necessários para manipular os dados. A operação desse bloco depende da configuração prévia de um documento XML que descreve cada um dos filtros usados e o tipo de dados de sensores que cada um permite tratar. Cada filtro configurado necessita de um ID, uma *routing key* de entrada e saída e uma classe correspondente ao tipo dados que irá receber e tratar. Depois de determinar os eventos da pessoa, os dados passam para o bloco *Sinker*, que serão todas as componentes que vão receber os dados processados e realizar o envio de notificações ou persistir os dados em FHIR.

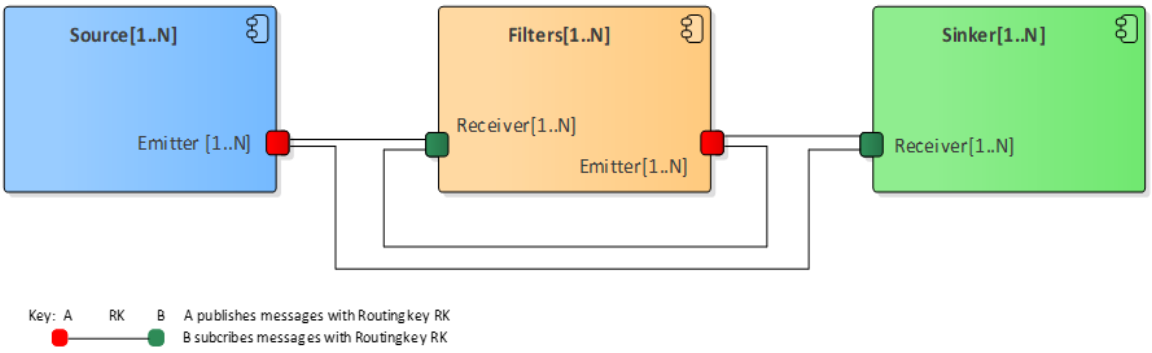

Figura 5.1 - Visão alto nível da arquitetura do sistema de deteção de eventos significativos.

5.1.1 Máquinas de estados

Para facilitar a implementação do projeto foram desenhadas máquinas de estados para representar os diferentes filtros usados no sistema de deteção de eventos significativos. Todas elas foram testadas a partir da ferramenta LTSA (*Labelled Transition System Analyser*) que verifica se a especificação de um sistema concorrente satisfaz as propriedades definidas para determinado comportamento [40] e comprova a inexistência de *deadlocks*. Os resultados desses testes podem ser consultados em anexo.

O filtro apresentado na Figura 5.2 permite detetar se uma pessoa está a descansar ou não, com recurso a apenas um sensor de pressão instalado na cama. O estado inicial desta máquina é "UNKNOWN", porque tal como acontece em todos os filtros que serão apresentados, nenhum dado foi recebido ainda, sendo impossível para o sistema conhecer o estado atual da pessoa. Depois de receber os primeiros dados do sensor da cama, o filtro passa para o estado INBED se o idoso estiver na cama ou NOTINBED, caso contrário. Se uma pessoa for para a cama, essa hora será gravada e comparada com a hora normal para descansar. Se estiver dentro do horário normal, o filtro mudará para o estado "*Resting*" e permanecerá assim enquanto o utilizador estiver deitado. Se a pessoa se levantar da cama, o próximo estado será "*Resting NotInBed*". Se esta situação ocorrer após o tempo esperado para acordar, passará para o estado "NOTINBED", e, caso contrário, permanecerá até ao horário esperado para acordar ou retornar ao estado "*Resting*" se a pessoa voltar para a cama. Se o estado "*Resting NotInBed*" durar mais que o esperado, deve ser emitido um alerta para o cuidador.

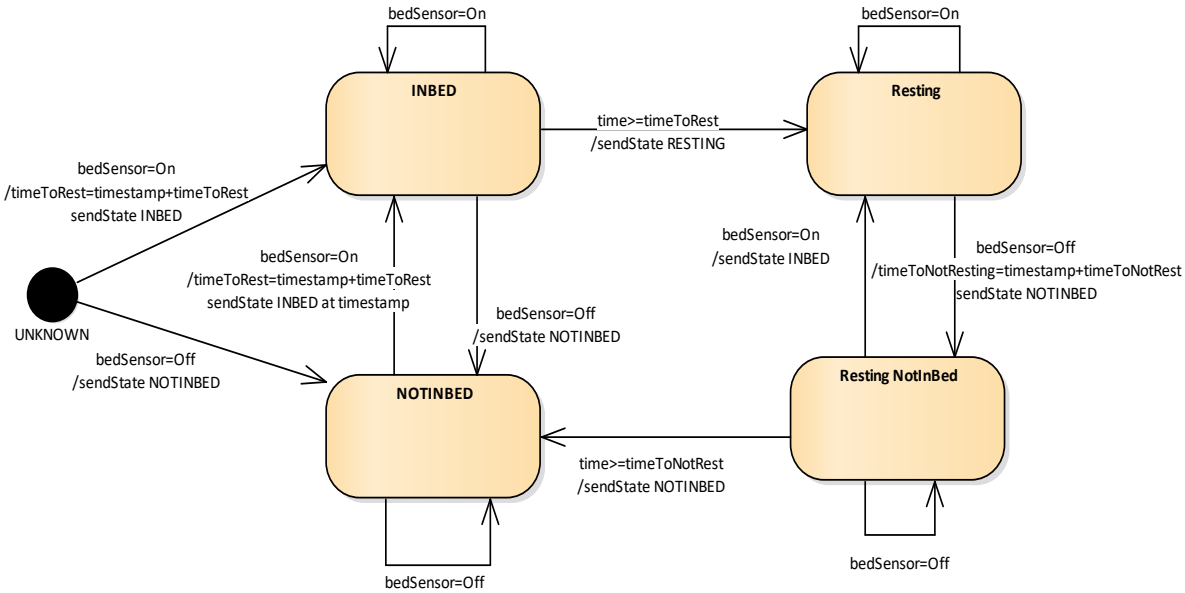

Figura 5.2 – Máquina de estados para detetar se um utilizador está a descansar**.**

A Figura 5.3 representa uma máquina de estados que também tem como objetivo detetar se o idoso está a descansar ou não. Porém, também é utilizado um sensor de contacto na janela. Desta forma, é possível detetar se a pessoa deixa definitivamente de descansar, porque abrir a janela do quarto será uma das primeiras atividades que irá realizar após se levantar.

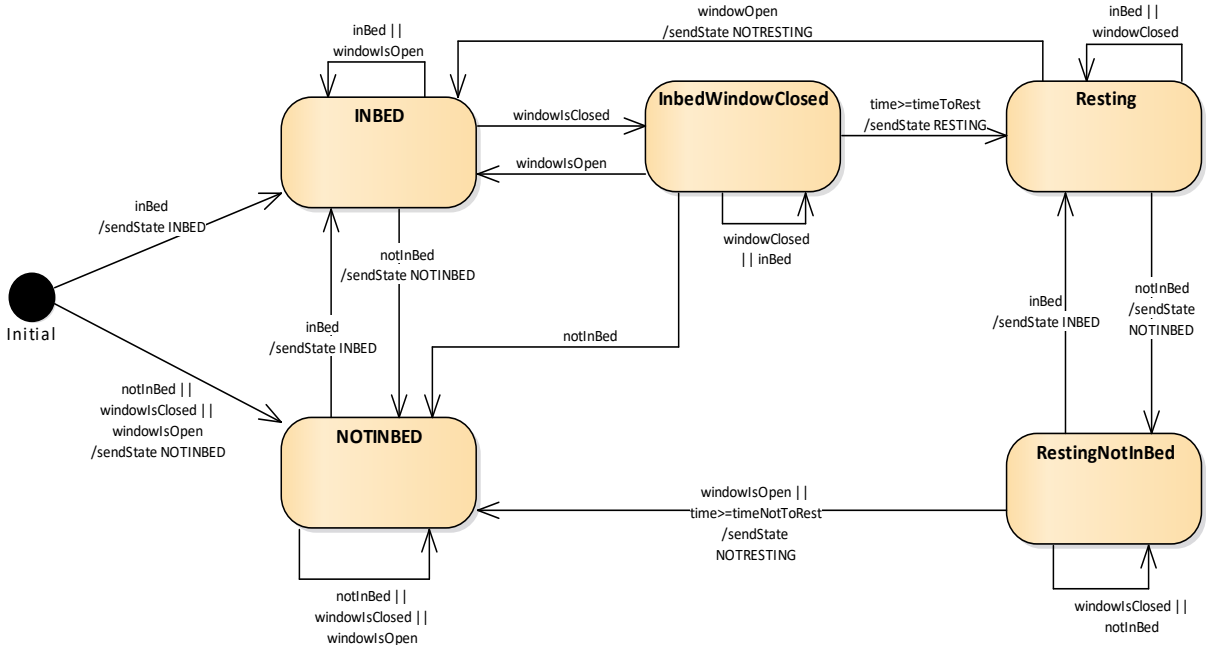

Figura 5.3 – Máquina de estados para detetar se um utilizador está a descansar com recurso a sensores na cama e na janela do quarto.

Outro dos casos de uso definidos anteriormente estava relacionado com o facto de detetar se a televisão estava ligada ou não, fazendo uso do sensor de contacto instalado na tomada da televisão e da box. Os únicos estados a enviar são os de televisão ligada, quando os valores dos sensores de consumo estão acima do limiar mais elevado, e desligada quando está abaixo desse limite. Estes estados, combinados com outros sensores, irão, mais tarde, permitir gerar alguns alertas. Futuramente, também poderá ser usado o estado *standby* para extrair outro tipo de estatísticas. A máquina de estados correspondente encontra-se representada na Figura 5.4.

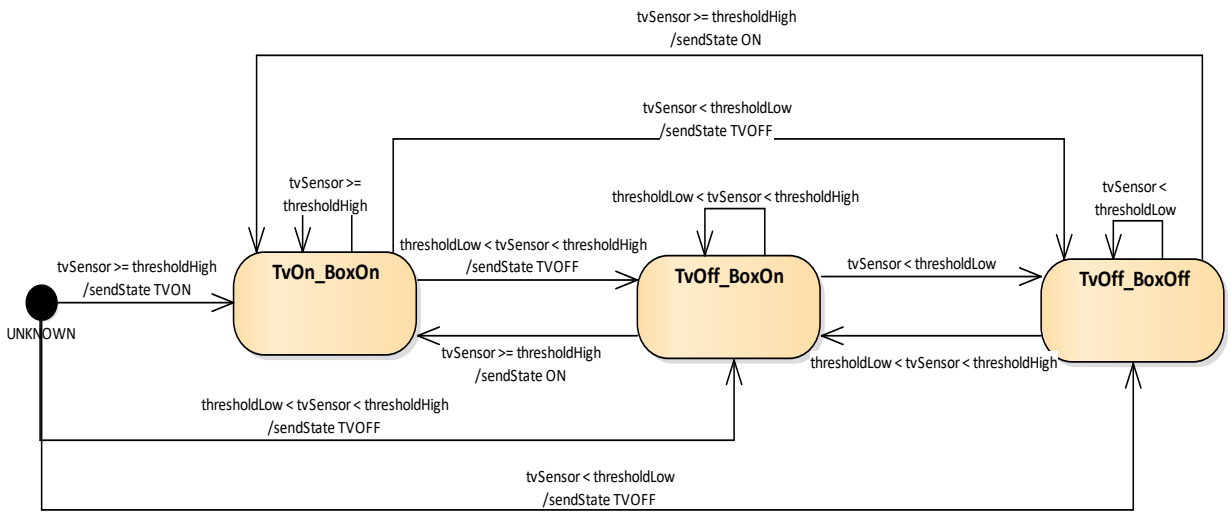

Figura 5.4 – Máquina de estados para detetar se um utilizador tem a televisão ligada.

Na Figura 5.5 está representada a máquina de estados que permite detetar se um utilizador está em casa ou não com recurso a um sensor de movimento instalado dentro do hall de entrada da casa e um a sensor de contacto instalado na porta de entrada. Ao iniciar o sistema, existem três possibilidades: o sensor de movimento deteta atividade, podendo indicar que o utilizador está dentro de casa; não deteta movimento, não existindo certeza se o utilizador está noutra divisão da casa ou se está fora de casa, ficando à espera de movimento (em casa) ou que a porta abra (a chegar a casa); ou a porta pode estar aberta, não sabendo se o utilizador está a entrar ou a sair de casa, passando ao estado *DoorOpen*. Só quando a porta fechar é que poderá passar para o estado *NotInHome*, se não detetar movimento dentro de casa ao fim de algum tempo, ou *InHall*, caso contrário.

Os processos de entrada e saída de casa são semelhantes. Após a abertura da porta passam a estados de porta aberta (*In* se estiver a entrar em casa e *Out* se estiver a sair) e após o fecho da porta. Caso não seja detetado movimento ao fim de algum tempo, isso significa que o utilizador não está em casa, e caso contrário é detetado movimento no hall e passa ao estado *InHall*. A partir daqui, caso não seja detetado movimento sem abrir a porta, transita então para o estado *NotInHall*.

Sempre que a porta estiver aberta durante um longo período de tempo deverá ser emitido um alerta de porta aberta.

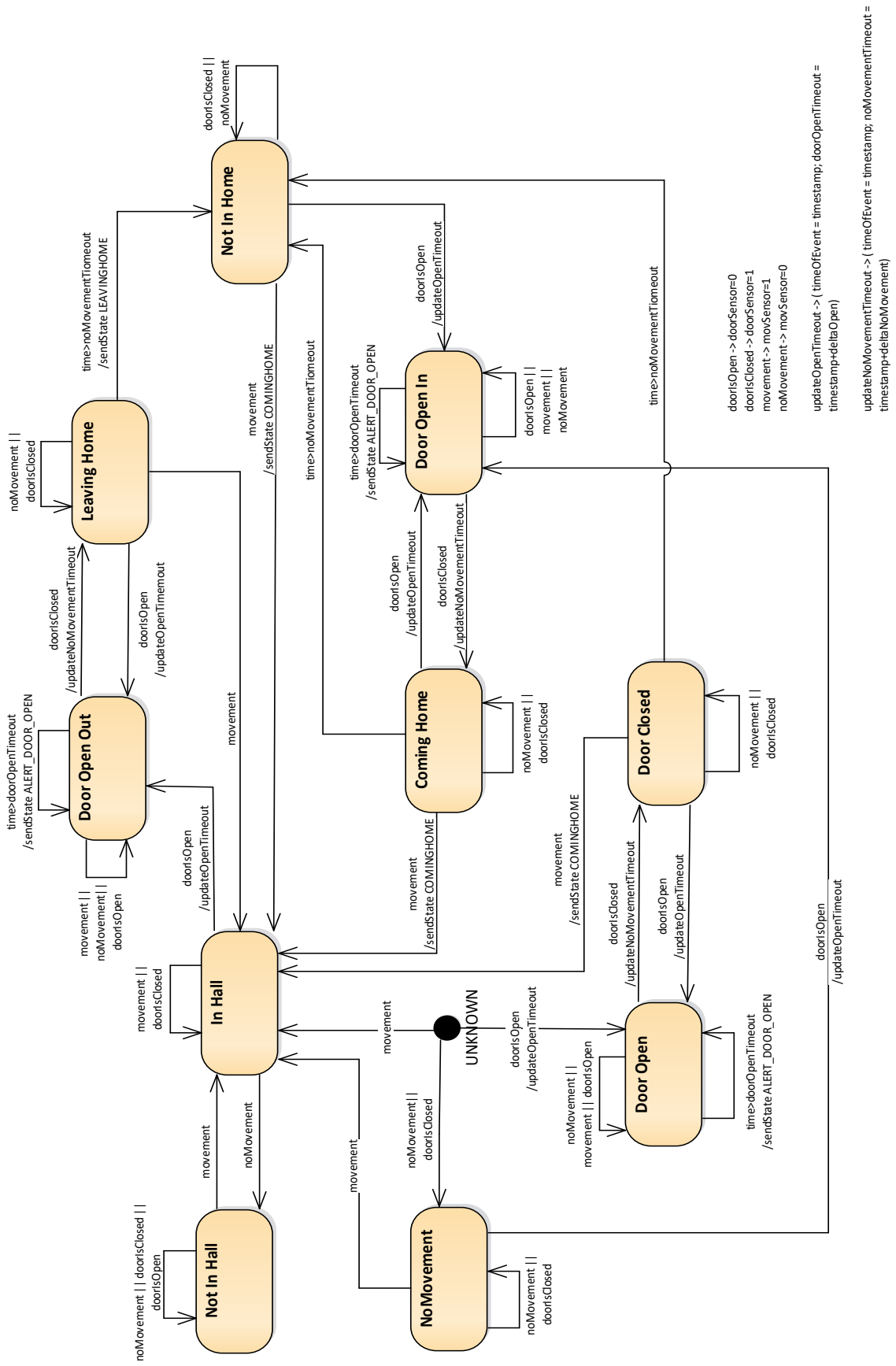

Figura 5.5 – Máquina de estados para detetar se um utilizador está em casa.

O diagrama anterior mostra a complexidade do sistema para a deteção deste evento. Pretende-se que todas as transições estejam representadas, mesmo que o dado do sensor a ser recebido não altere o estado atual da máquina. Porém, durante a implementação do sistema este processo é simplificado, sendo apenas necessário ter em conta os dados que podem levar a uma mudança de estado ou a gerar alertas. A Figura 5.6 mostra que para o estado *Door Open,* apenas foi necessário ter em conta a receção do valor de porta fechada, para passar a *Door Closed* e ao alarme caso a porta fique aberta durante muito tempo, apesar de se manter no mesmo estado.

```
public  RabbitMessage reactDoorOpen(SensorValue sensorValueIn) {
    RabbitMessage res = null;
    if( sensorValueIn.getSensorId().equalsIgnoreCase(doorSensorId) && sensorValueIn.getValue() == CLOSED) {
        state = DOOR_CLOSED;
        timeToLeaveHome = sensorValueIn.getTimestamp().getTime()+ deltaMovement;

    }else if( sensorValueIn.getSensorId().equalsIgnoreCase(clockId) && sensorValueIn.getTimestamp().getTime() > timeToCloseDoor) {
        res =sendDoorOpenAlert(sensorValueIn.getTimestamp());
    }
    return res;
}
```

Figura 5.6 – Exemplo de código Java relativo ao estado *Door Open.*

Neste processo também foram desenhadas as máquinas de estados relacionadas com os alertas que vão ser gerados. Na Figura 5.7 são gerados dois tipos de alertas: um quando o utilizador não acorda algum tempo após a hora normal de se levantar e outro quando o utilizador ainda não foi para a cama algum tempo após a hora habitual. Existe um tempo adicional *delta,* previamente definido, que é incrementado ao tempo esperado para um utilizador se deitar/acordar cada vez que um alerta é lançado, para que mais tarde seja possível voltar a enviar outra notificação para o cuidador. Na Figura 5.8 o processo é semelhante, com o alerta a ser enviado estar relacionado com o facto de um utente estar ou não em casa à hora habitual.

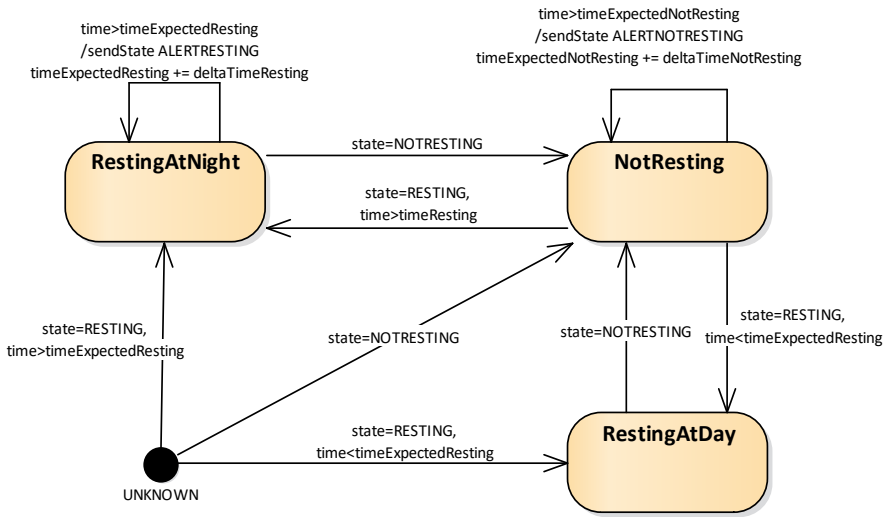

Figura 5.7 – Máquina de estados de alertas relacionados com o descanso do utilizador.

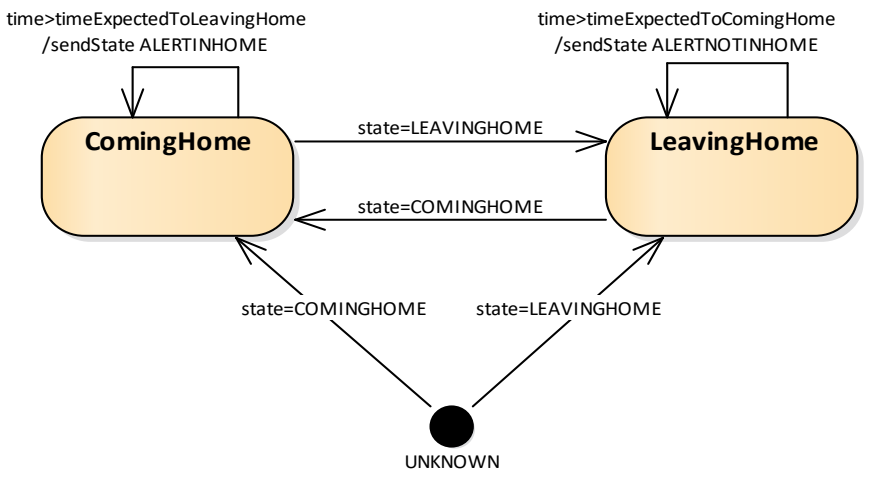

Figura 5.8 – Máquina de estados de alertas relacionados com a presença de utente em casa

A Figura 5.8 mostra a máquina de estados que conjuga os estados de presença/ausência do utente da sua habitação com o estado da televisão. Os alertas aqui gerados, para além dos referidos anteriormente, incluem um que avisa o cuidador quando o idoso sai de casa e a televisão fica ligada.

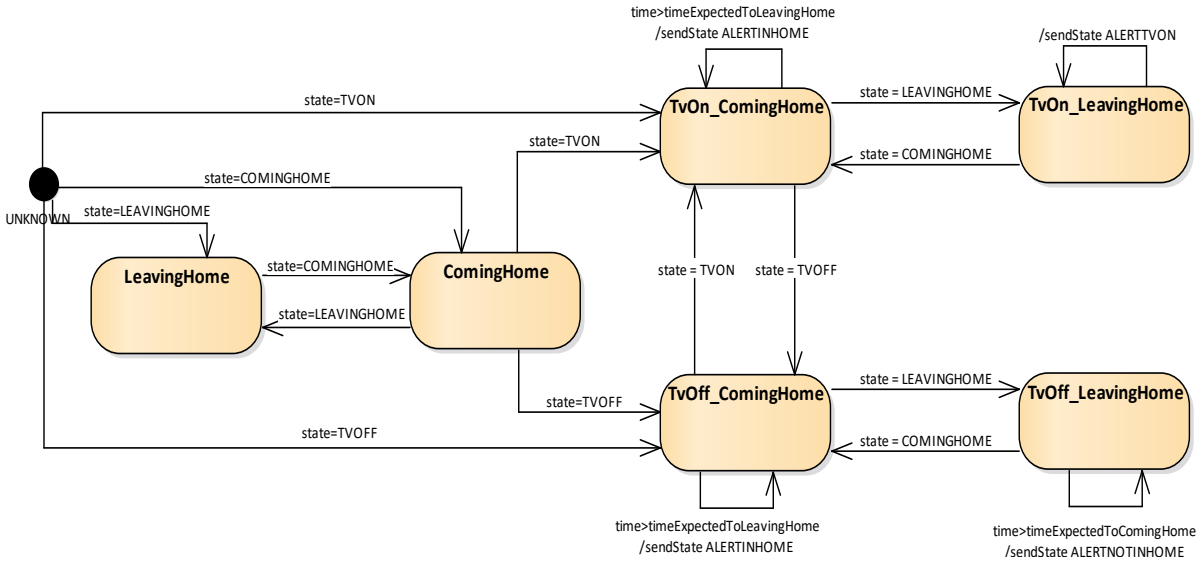

Figura 5.9 – Máquina de estados de alertas relacionados com a presença de utente em casa e a televisão.

5.2.2 Filtros

Como explicado anteriormente neste capítulo, os filtros têm um papel essencial na arquitetura do sistema. Os filtros necessários para o funcionamento do sistema de deteção de eventos significativos são apresentados de seguida e de forma resumida:

- ***Low Pass*** – Este filtro será responsável por selecionar apenas os dados do sensor que sejam relevantes para a deteção de um determinado evento da pessoa.
- ***Person Check*** – Este filtro será responsável por verificar, periodicamente, o estado de uma pessoa. Isso permite atualizar informações sobre uma pessoa ou gerar alertas para os cuidadores.
- ***Sensor State*** – Este filtro gera o estado atual do sensor, com base na análise dos últimos cinco valores recebidos.
- ***Resting State*** – Com base no estado atual da pessoa e nos valores recebidos dos sensores num determinado instante, este filtro vai detetar se uma pessoa está a descansar ou não.
- ***Home State*** – Com base no estado atual da pessoa e nos valores recebidos dos sensores num determinado instante, este filtro vai detetar se uma pessoa está a em casa ou não.
- ***Event Stats*** – Com base no *timestamp* de cada evento detetado vai calcular a média total e os desvios padrão de todos os eventos recebidos, e também fará esse cálculo para cada dia da semana.
- ***Person Events*** – Com base nas informações enviadas pelos filtros *Person Check*, *Resting State* e *Event Stats* será gerado o evento detetado e os registos do idoso vão ser atualizados.

Para entender a eficácia dos filtros desenvolvidos para esta componente serão utilizadas matrizes de comparação, representadas a verde na figura seguinte como *Person Events CM*. Desta forma, será possível acompanhar o desempenho do algoritmo, informando o número de eventos detetados, não detetados e não esperados, que correspondem respetivamente aos verdadeiros positivos, falsos negativos e falsos positivos. No projeto SOCIAL, o tratamento de falsas medições será feita com recurso ao *feedback* dos cuidadores sobre os eventos notificados e o treino do algoritmo por aprendizagem, como já foi referido anteriormente.

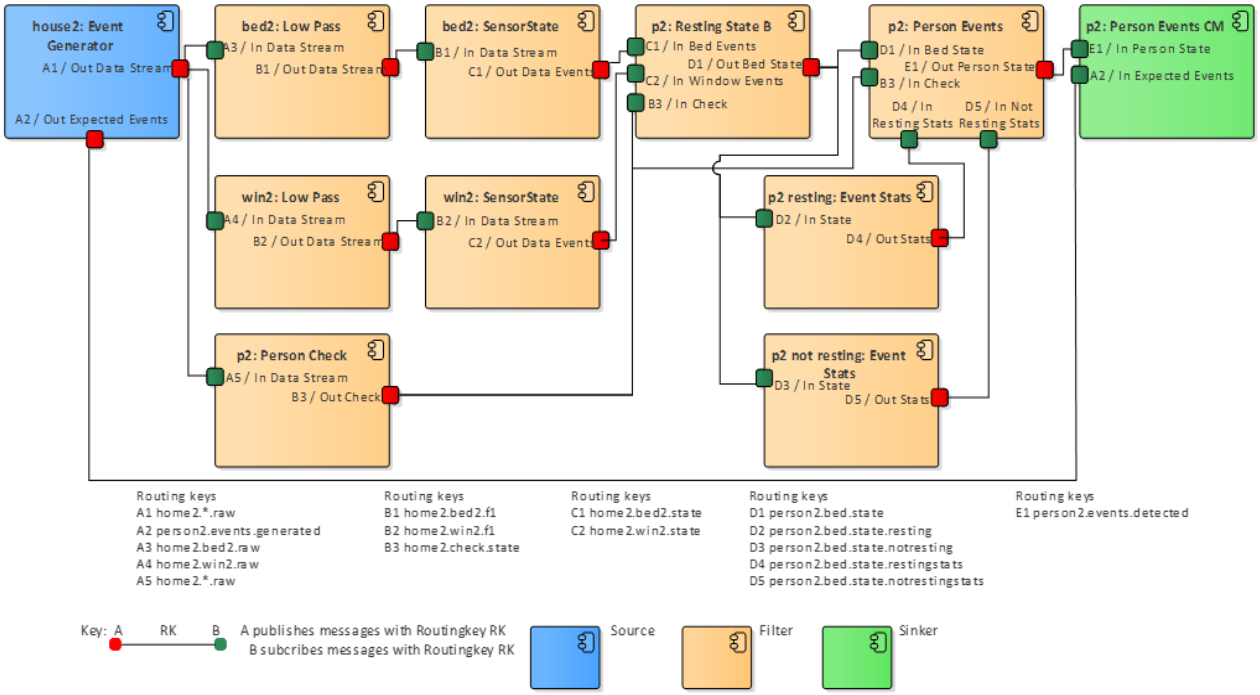

Figura 5.10 – Plano sequencial de filtros para detetar se o idoso está a descansar

Capítulo 6
Implementação

6.1 Gerador de eventos

À data da implementação da solução, não foi possível reunir dados reais de sensores para os casos de estudo que se pretendia analisar. Para garantir a realização do plano para o estágio, foi criado um gerador de dados de sensores. Com base num documento de configuração, com os dados sobre os hábitos de cada pessoa e o conjunto de sensores presentes numa determinada casa, é possível gerar dados para cada sensor num determinado período temporal da simulação e simular rapidamente situações anómalas que, em ambiente real, poderiam demorar muito tempo a acontecer.

As configurações da simulação são carregadas a partir de um ficheiro XML, que facilita a criação de diferentes simulações e a alteração das mesmas. A Figura 6.1 representa um exemplo de um ficheiro de configuração da simulação. Cada ficheiro contém os seguintes elementos:

- **Simulação**: Onde são definidas as datas de início e de fim da simulação e a frequência com que serão gerados os dados da mesma.
- **Casa**: Apresenta apenas um atributo identificador da habitação.
- **Sensores**: Reúne todos os sensores presentes numa casa, sendo que cada sensor é identificado pelo seu tipo, o local onde está instalado, um identificador e um nome. Além disso, poderão ser adicionadas falhas de forma a simular anomalias nos equipamentos.
- **Pessoas**: Este elemento reúne todas as pessoas que habitam numa casa. Cada pessoa tem com atributos um nome, um identificador e uma hora mínima de deitar. A acrescentar a isso, cada elemento possui um conjunto de configurações que servirão de base para a criação de dados ao longo da simulação. Incluem-se as atividades do utilizador, como deitar, levantar ou abrir a porta que podem ser agendadas para um dia específico, um dia da semana, um dia do mês, algo que acontece todos os dias ou a hora específica a que ocorre o evento.

```xml
<?xml version="1.0" encoding="UTF-8"?>

<simulation start="2018-04-01 00:00:00" end="2018-04-02 23:59:00" increment="60000" pace="1">

    <house id="house1">
        <sensors>
            <sensor type="1" local="1" id="cam1" name="cama">
                <anomalies>
                    <anomaly type="switch" pattern=".*:.5:.*"/>
                </anomalies>
            </sensor>
            <sensor type="2" local="1" id="jan1" name="janela quarto"></sensor>
        </sensors>

        <persons>

            <person name="Maria" id="person1" mintimesleep="20:00">
                <personconfigs levels="2">
                    <personconfig           type= "resting" base="1"  time="00:01"></personconfig>
                    <personconfig           type= "inbed" base="1"  time="00:01"></personconfig>

                    <personconfig           type= "notresting" base="1"  time="08:00"></personconfig>
                    <personconfig           type= "notinbed" base="1"  time="08:00"></personconfig>

                    <personconfig level="1" type= "notinbed" base="1"  time="03:00"></personconfig>
                    <personconfig level="1" type= "inbed" base="1"  time="03:10"></personconfig>

                    <personconfig level="2" type= "notinbed" base="1"  time="05:00"></personconfig>
                    <personconfig level="2" type= "inbed" base="1"  time="05:10"></personconfig>

                </personconfigs>
            </person>
        </persons>

    </house>
</simulation>
```

Figura 6.1 – Exemplo de ficheiro de configuração de uma simulação

A partir da análise das configurações será possível detetar, para todos os dias da simulação, os eventos que vão ocorrer com uma determinada pessoa. Por exemplo, se nas configurações estiverem presentes apenas a hora de deitar e de levantar da pessoa para trinta dias de simulação, então vão ser gerados sessenta eventos relativos àquela pessoa.

Com os eventos das pessoas, que foram gerados anteriormente, também será possível detetar eventos de sensores. Estes eventos correspondem às alterações de estado, num dado momento, de um determinado sensor. Por exemplo, se uma pessoa se deitar às 22h, então haverá um sensor de pressão na cama que irá alterar o seu estado de 0 para 1.

Após a deteção de todos os eventos, será possível gerar dados com a frequência indicada no ficheiro de configuração. O gerador irá criar dados com base nas informações existentes para cada pessoa. Quando encontra um novo evento, os estados dos sensores respetivos serão alterados e vão ser prolongados no tempo até ser encontrado outro evento. Usando o último exemplo, se existir um evento de sensores onde o sensor de pressão da cama passe a 1 às 22h, então esse valor será mantido por todas as frações de tempo geradas até à hora em

que existir um novo evento em que esse sensor passe a ter o valor 0, que irá coincidir com a hora de levantar. Porém, também foi inserido algum ruído nos dados enviados, para que estes não estivessem completamente puros. Desta forma, foi possível simular pequenas falhas momentâneas na receção dos dados, que seriam mais tarde resolvidas na fase de tratamento dos dados.

6.2 Processamento dos dados

Imediatamente após a geração dos dados é realizado o envio dos mesmos para um tópico, com recurso ao RabbitMQ, permitindo que uma mensagem possa ser enviada para mais do que um subscritor, ao contrário das *queues*. Para o processo de envio, cada mensagem leva informação sobre o identificador do sensor, o seu estado e a data em que foi recolhida esta informação. Além disso, é adicionada uma chave de encaminhamento que vai ajudar, mais tarde, na identificação e junção de informação sobre o sensor e facilitar o processamento dos dados, apresentando o seguinte formato "Casa.SensorID.FaseDoProcessamento". A Figura 6.2 mostra um exemplo de configuração de filtros para a casa *house1*, equipada com um sensor de pressão na cama e um sensor de contacto na janela do quarto.

```
<filters>
<filter id="pbcam1" routingkeyin="house1.cam1.raw"  routingkeyout="house1.cam1.f1"      classname="SensorFilter"/>
<filter id="pbcam2" routingkeyin="house1.jan1.raw"  routingkeyout="house1.jan.f1"       classname="SensorFilter"/>

<filter id="h1clock" routingkeyin="house1.cam1.raw" routingkeyout="house1.check.state"  classname="PersonCheck"/>

<filter id="stcam1" routingkeyin="house1.cam1.f1"   routingkeyout="house1.cam1.state"   classname="SensorState"/>
<filter id="stjan1" routingkeyin="house1.jan1.f1"   routingkeyout="house1.jan1.state"   classname="SensorState"/>
```

Figura 6.2 – Exemplo de ficheiro XML de configuração de filtros

Para cada uma das fases de processamento são criadas *threads* para cada sensor existente na simulação multiplicadas pelo número de fases do processamento. Isto irá permitir que cada canal que se abre fique disponível para tratar exclusivamente os dados em qualquer altura, sem criar confusão entre os dados.

Na primeira fase, os dados de um determinado sensor serão identificados pela sua chave e consumidos para avaliar o estado do sensor. De acordo com um estudo de Sharma *et al* [41], existem quatro tipos de métodos para identificar falhas: métodos baseados em regras, métodos de estimativa, métodos baseados em aprendizagem e métodos baseados em análise de séries temporais. O último enumerado foi usado para manipular os dados recebidos. Para tal são tidos em consideração os últimos cinco estados recebidos daquele sensor, onde a maioria ditará qual o seu verdadeiro estado atual. Desta forma são evitados alguns falsos

positivos, originados por alterações momentâneas do estado dos sensores. Por exemplo, caso um utente esteja deitado, mas seja detetado apenas um estado do sensor de pressão igual a 0, não será indicado de imediato que ele se levantou. Após ser identificado o valor do sensor, este será enviado para o tópico, mas com uma chave diferente, relativa à fase de processamento em que se encontra.

Na segunda fase, o consumo dos dados do tópico é feito de forma semelhante. Porém, com base nos valores obtidos anteriormente, podem ser identificados eventos do sensor e alterar em definitivo o estado do sensor. Essa informação será adicionada ao tópico, com uma nova chave, para que seja usada posteriormente.

Por fim, com a informação anterior, será possível detetar os eventos da pessoa, como deitar ou levantar da cama. Esta informação será enviada mais uma vez para o tópico, mas desta vez com a chave no formato adequado a um evento da pessoa. Desta forma, a informação poderá ser usada para a aprendizagem das rotinas de um utente e identificar eventos significativos, sendo ainda possível ser consumida por mais do que um subscritor e usada para gerar mais do que um tipo de alerta.

6.3 Execução do sistema

Existem três modos diferentes de simulação: um modo *default*, onde se definem as rotinas do utilizador para uma determinada tarefa e para todos os dias da simulação, outro onde é possível especificar as tarefas por dias e dias da semana e, por fim, um modo aleatório, onde é possível definir um intervalo em que um evento pode ocorrer. Isto permitirá que um determinado evento não ocorra sempre à mesma hora e conferir à simulação um maior realismo. Porém, os dados recebidos pelos sensores reais podem, por vezes, apresentar algum ruído. Por isso, também existem três modos diferentes para gerar ruído: um modo sem qualquer ruído nos dados, outro onde é possível introduzir ruído periódico ou então gerar ruído aleatório e conferir ainda mais realidade à simulação.

Foi implementada também uma página HTML que apresenta a matriz de comparação já referida no capítulo anterior. Esta lê, em tempo real, um ficheiro JSON com os eventos detetados, não detetados e não esperados. Na Figura 6.3 é possível observar duas tabelas: a da esquerda apresenta a lista de todos os eventos, diferenciados por cores e com a indicação da data esperada para ocorrer e a altura em que foi detetado, e a tabela da direita com a matriz de comparação, com os quatros estados possíveis e o número de eventos detetados,

não detetados e não esperados. Além disso, é possível selecionar o tipo de eventos que queremos observar.

Social Simulator

Timeline												
32%												
House	Resting			Not Resting			In Bed			Not in bed		
	Detected	Not Det.	Not exp.	Detected	Not Det.	Not exp.	Detected	Not Det.	Not exp.	Detected	Not Det.	Not exp.
House1 (Sem ruído)	10	0	1	10	0	1	30	0	0	30	0	0
House2 (10% min 5)	10	0	2	10	0	2	30	0	0	30	0	0
House3 (10% min 3)	10	0	2	10	0	2	30	0	0	30	0	0
House4 (20% min 3 e 7)	10	0	1	10	0	1	30	0	0	30	0	0
House5 (40% min 2,3,4,7)	0	10	8	0	10	8	21	9	1440	30	0	1431
House6 (10% dia 6)	8	2	5	9	1	4	26	4	3	27	3	2

Figura 6.3 – Página web com a listagem dos eventos do idoso

Capítulo 7
Testes e Resultados

Neste capítulo são apresentados os principais resultados da execução da componente de deteção de eventos significativos e alguns dos testes realizados.

Relativamente à abordagem escolhida para os testes optou-se pela *Black Box Testing*. Foram realizados testes funcionais com base nos requisitos funcionais definidos para este estágio, definindo os dados de entrada para cada teste e o resultado esperado, para que fosse possível validar cada caso. Além disso, foram executados testes de desempenho, de forma a perceber como reagia o sistema ao executar uma simulação com várias casas, obrigando o sistema a gerar e a ler mais dados de sensores por minuto de ambiente simulado.

7.1 Detetar estado de descanso com sensor de cama

As experiências iniciais foram projetadas para validar que o sistema identifica corretamente um conjunto reduzido de comportamentos do utilizador final (*Resting, Not Resting, In Bed, Not In Bed*) com base em dados de um sensor de contacto na cama e no filtro correspondente, descrito anteriormente. A simulação foi executada por um período de dez dias, onde a cada 5 milissegundos eram gerados dados de sensores para um minuto da simulação. Para ilustrar os resultados, optou-se por relatar seis execuções. Uma execução sem ruído (A), cinco com ruído gerado, com alteração do estado esperado, a cada minuto terminado em 5 (B), a cada minuto terminado em 3 (C), a cada minuto terminado em 3 ou 7 (D), a cada minuto terminado em 3, 4, 5 ou 7 (E) e ruído para um dia específico (F). Os resultados foram os seguintes:

Tabela 7.1 – Resultados da execução para deteção do estado de descanso de um utilizador

Evento	Estado	A	B	C	D	E	F
Resting	Detected	10	10	10	10	0	8
	Not detected	0	0	0	0	10	2
	Not expected	0	0	0	0	8	1
Not resting	Detected	10	10	10	10	0	9
	Not detected	0	0	0	0	10	1
	Not expected	0	0	0	0	8	0
In bed	Detected	30	30	30	30	21	26
	Not detected	0	0	0	0	9	4
	Not expected	0	0	0	0	1440	3
Not in bed	Detected	30	30	30	30	30	27
	Not detected	0	0	0	0	0	3
	Not expected	0	0	0	0	1431	2

Legenda:

A - Sem ruído
B - Ruído a cada minuto terminado em 5
C - Ruído a cada minuto terminado em 3
D - Ruído a cada minuto terminado em 3 ou 7
E - Ruído a cada minuto terminado em 3,4,5 ou 7
F - Ruído num dia específico

Após a execução da simulação é possível observar que, para a casa que não apresentava qualquer ruído no sensor de pressão da cama, todos os eventos previstos foram detetados pelo sistema de deteção de eventos. Além disso, para as casas B, C e D, onde foi adicionado até 20% de ruído no sensor, todos os eventos previstos também foram detetados. Este facto mostra que o tratamento dos últimos cinco dados enviados pelo sensor está a ser realizado de forma correta. Para o caso E, com a introdução de 40% de ruído, alguns eventos foram detetados por não existir ruído entre os minutos acabados em 8 e 2, o que faz com que os eventos esperados para os minutos terminados em 0 sejam detetados. Porém, como há muito ruído nos restantes minutos, são detetados muitos eventos não esperados, principalmente para os casos *inbed* e *notinbed*. Por fim, para a casa F, todos os eventos previstos, para os dias sem ruído no sensor, foram detetados, algo que não aconteceu num dos dias, onde essa alteração não permitiu que o sistema detetasse eventos.

7.2 Detetar estado de descanso com sensor de cama e de janela

Nesta experiência, as condições de simulação são semelhantes, mas a deteção do evento já é feita com recurso a dados de um sensor de pressão na cama e de contacto na janela do quarto. Tal como no caso anterior, foi executada uma simulação por um período de dez dias, onde a cada 5 milissegundos eram gerados dados de sensores para um minuto da simulação.

Tabela 7.2 – Resultados da execução para deteção do estado de descanso de um utilizador, com dados de um sensor de contacto na janela e um de pressão na cama durante 10 dias

Event	State	A	B	C	D	E	F
Resting	Detected	10	10	10	10	0	8
	Not detected	0	0	0	0	10	2
	Not expected	0	2	0	0	226	2
Not resting	Detected	10	10	10	10	1	9
	Not detected	0	0	0	0	9	1
	Not expected	0	2	0	0	225	1
In bed	Detected	30	30	30	30	21	26
	Not detected	0	0	0	0	9	4
	Not expected	0	0	0	0	1402	3
Not in bed	Detected	30	30	30	30	30	27
	Not detected	0	0	0	0	0	3
	Not expected	0	0	0	0	1392	2

Legenda:

A - Sem ruído
B - Ruído a cada minuto terminado em 5
C - Ruído a cada minuto terminado em 3
D - Ruído a cada minuto terminado em 3 ou 7
E - Ruído a cada minuto terminado em 3,4,5 ou 7
F - Ruído num dia específico

A execução desta simulação mostrou resultados bastante semelhantes aos da realizada anteriormente. Para as casas A, C e D foram detetados todos os eventos que eram esperados no início da simulação. O caso B também detetou todos os eventos esperados inicialmente, porém, para os eventos *Resting* e *Not Resting* foram encontrados dois estados em cada um que não eram esperados. Este resultado verificou-se não só devido ao facto desta simulação exigir muita capacidade computacional de processamento, com tempos próximos para a geração de dados, e pela execução numa máquina com recursos partilhados. Estes eventos não esperados verificam-se de novo para a casa E, em que a causa da ocorrência é a mesma. Esse facto não interferiu com a execução esperada para a casa F.

7.3 Detetar entrada e saída de casa

Para esta experiência, foi realizada uma simulação com duas casas, cada uma com um sensor de movimento no hall de entrada e um sensor de contacto na porta. Foram simulados dez dias, com cada um dos utilizadores a sair de casa às 13h e a regressar às 14h, porém o segundo passa pelo hall de entrada várias vezes, mas sem sair de casa. Na tabela 7.3 são apresentados os resultados obtidos.

Tabela 7.3 – Resultados da execução para deteção de saída e entrada em casa de um utente.

Event	State	A	B
Go outside	Detected	10	10
	Not detected	0	0
	Not expected	0	0
Come inside	Detected	10	10
	Not detected	0	0
	Not expected	0	0

Legenda:

A – Utente sai e entra em casa uma vez por dia

B - Utente sai e entra em casa uma vez por dia e acorda duas vezes a meio da noite e passa no hall de entrada

Os resultados apresentados estão de acordo com aquilo que se esperava, com a totalidade dos eventos previstos a serem detetados pelo sistema de deteção de eventos significativos.

Por fim, foi realizada uma simulação composta pelos sensores utilizados nas execuções anteriores. Os principais objetivos passavam por observar de que forma reagia o sistema a uma carga mais elevada de trabalho e criar um cenário mais dinâmico, com mais ações a acontecer. Para todas as casas foram simulados 10 dias de eventos, tendo as casas A e B apenas sensores instalados na cama e na janela, para detetar se o utente está a descansar, as casa C e D contém sensores na porta e de movimento na entrada para saber se o utilizador sai de casa e se está no hall de entrada, e as casas E e F conjugam todos os sensores anteriores.

Os resultados apresentados na tabela 7.4 mostram que para as casas A, B, C e D todos os eventos previstos foram detetados sem qualquer problema. No entanto, para as casas E e F existiu um evento não detetado em quase todos os eventos avaliados. A carga de processamento para esta simulação era elevada, o que poderá ter afetado a performance no sistema na fase inicial. Apesar de tudo é importante relembrar que estes testes foram realizados na máquina onde foi implementado o código da solução, com outros processos a executar em paralelo.

Tabela 7.4 – Resultados da execução da simulação para deteção de estado de descanso e de eventos de saída e entrada em casa de um utente.

Event	State	A	B	C	D	E	F
Resting	Detected	10	10	-	-	9	9
	Not detected	0	0	-	-	1	1
	Not expected	0	0	-	-	0	0
Not resting	Detected	10	10	-	-	9	9
	Not detected	0	0	-	-	1	1
	Not expected	0	0	-	-	0	0
In bed	Detected	10	30	-	-	10	29
	Not detected	0	0	-	-	0	1
	Not expected	0	0	-	-	0	0
Not in bed	Detected	10	30	-	-	10	29
	Not detected	0	0	-	-	0	1
	Not expected	0	0	-	-	0	0
Coming home	Detected	-	-	10	10	9	9
	Not detected	-	-	0	0	1	1
	Not expected	-	-	0	0	0	0
Leaving home	Detected	-	-	10	10	9	9
	Not detected	-	-	0	0	1	1
	Not expected	-	-	0	0	0	0
In hall	Detected	-	-	10	30	9	29
	Not detected	-	-	0	0	1	1
	Not expected	-	-	0	0	0	0
Not in hall	Detected	-	-	10	30	9	29
	Not detected	-	-	0	0	1	1
	Not expected	-	-	0	0	0	0

Legenda:

A, C e E – Utente levanta-se e deita-se uma vez por dia

B, D e F – Utente acorda duas vezes a meio da noite e passa no hall de entrada

7.4 Testes funcionais

Os testes funcionais permitiram avaliar cada uma das funcionalidades desenvolvidas e detetar se os resultados obtidos, após a introdução de variáveis previamente definidas, coincidiam com os resultados esperados. Nesse sentido, foram definidos alguns testes, cujos *inputs* se baseavam em grande parte nas personas e cenários definidos no Capítulo 3 e nos casos de uso que foram trabalhados ao longo do estágio. Os resultados destes testes são apresentados na Tabela 7.5.

Tabela 7.5 – Resultados dos testes funcionais do sistema.

ID	Evento	Evento significativo	Resultado esperado	Resultado Obtido
T1	O utente não se levantou	O utente não se levantou até às 9:30h	O comportamento habitual não é detetado	Igual ao esperado
T2	O utente saiu de casa meio da tarde	O utente sai de casa às 15h	É gerado um evento não esperado	Igual ao esperado
T3	O utente não abriu a janela	O utente não abriu a janela até às 9h	O comportamento habitual não é detetado	Igual ao esperado
T4	O utente não se deitou na cama	Às 22:30h o utente ainda não foi para a cama	O comportamento habitual não é detetado	Igual ao esperado
T5	O utente levantou-se a meio da noite	O utente levanta-se a meio da noite entre as 3h e as 3:10h	É gerado um evento não esperado	Igual ao esperado
T6	O utente saiu de casa a meio da noite	O utente saiu de casa às 4h	É gerado um evento não esperado	Igual ao esperado
T7	O utente não fechou a janela	Às 21h o utente ainda não fechou a janela	O comportamento habitual não é detetado	Igual ao esperado
T8	O utente deixou a porta aberta	O utente abriu a porta às 15h e às 15:15h ainda não tinha fechado	É gerado um evento não esperado	Igual ao esperado

Capítulo 8
Gestão do trabalho

Na base do sucesso de um projeto está a capacidade de preparar e organizar o plano de trabalho futuro. Para isso, é necessário determinar as tarefas que deverão ser executadas, estimar quanto tempo irão demorar, encontrar as ameaças que podem atrasar ou colocar um fim ao desenvolvimento do projeto e, por fim, encontrar estratégias para colmatar esses possíveis problemas.

Neste capítulo é apresentado o modelo de desenvolvimento adotado, o plano de trabalho que foi definido para todo o ano, os dados do esforço despendido ao longo do primeiro semestre e os riscos associados à componente de deteção de eventos significativos desenvolvida durante o estágio.

8.1 Metodologia

O modelo utilizado ao longo de todo o projeto será o ACDM (*Architecture Centric Design Model)* [37]. Este método dá suporte ao desenho da arquitetura, aperfeiçoando-a até ser atingido um estado de certeza que permita avançar para a implementação e desenvolvimento do projeto.

O ACDM pode ser dividido em duas partes. A primeira é de incerteza em relação à arquitetura, onde são recolhidos os recursos funcionais, os atributos funcionais e as restrições, para além da definição da arquitetura e realização de experiências até ser confirmada. Na segunda parte, já com a certeza da arquitetura, é possível avançar para a implementação do projeto.

A figura seguinte mostra as diferentes fases do modelo ACDM, mostrando a ordem pela qual cada uma ocorre e a sua divisão por quatro etapas: requisitos, desenho, experimentação e produção.

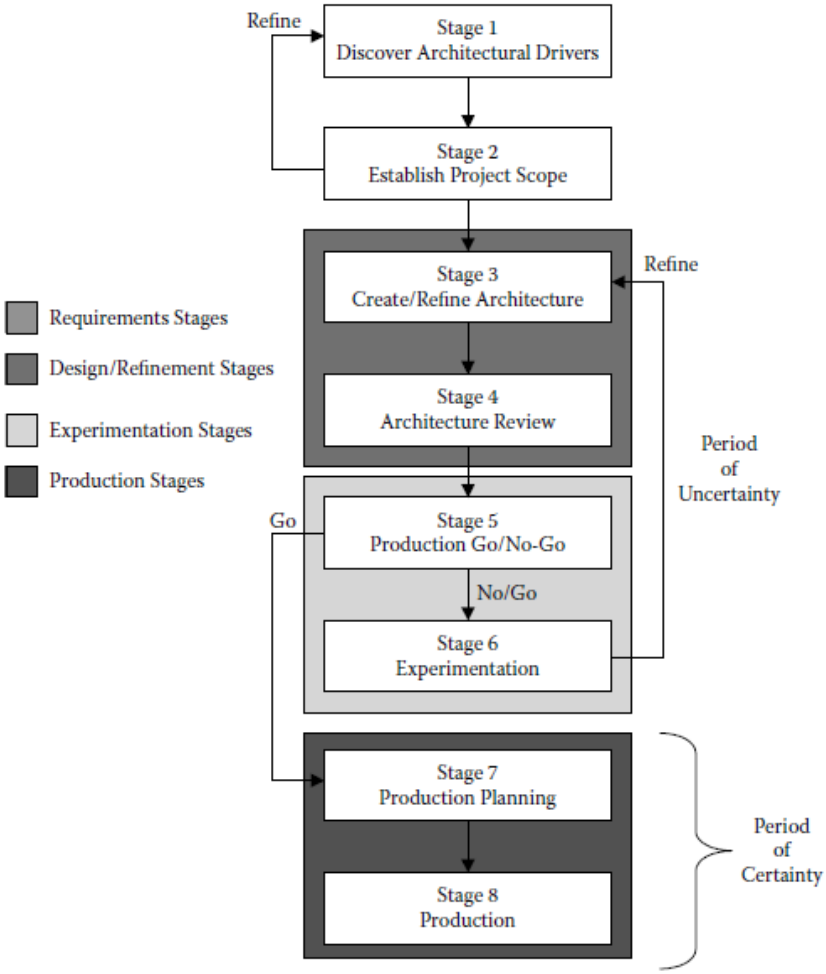

Figura 8.1 - Representação por fases do modelo ACDM [37].

O modelo ACDM é constituído por oito fases divididas da seguinte forma:

- **Fase 1: Descoberta de *drivers* arquiteturais** - Fase dedicada à descoberta e documentação dos requisitos funcionais, dos atributos de qualidade e das restrições do projeto.
- **Fase 2: Estabelecimento do âmbito do projeto** - Nesta fase são analisados os documentos criados anteriormente e definidos os requisitos prioritários, ajudando assim na definição dos objetivos do projeto.
- **Fase 3: Criação/Refinamento da arquitetura** - Entrando na etapa de desenho, esta fase poderá surgir após a fase dois, onde será criada uma arquitetura de raiz, capaz de respeitar os requisitos definidos, ou então depois da fase seis, caso as experiências realizadas mostrem a necessidade de alterar a arquitetura.

- **Fase 4: Revisão da arquitetura** - Esta fase é fundamental em todo o processo, porque é aqui que se reflete sobre possíveis problemas da arquitetura e é feita uma avaliação dos riscos.
- **Fase 5: Decisão de produção** - Nesta fase é tomada a decisão de avançar para o projeto, caso não existam dúvidas sobre a arquitetura ou, caso contrário, passar para a fase de experimentação.
- **Fase 6: Experimentação** - Fase dedicada a experiências relacionadas com a arquitetura, de modo a mitigar riscos e resolver possíveis problemas.
- **Fase 7: Planeamento de produção** - Já com certezas da arquitetura, é realizada uma descrição detalhada dos elementos a desenvolver na fase de produção.
- **Fase 8: Produção** - Fase dedicada à produção do projeto.

8.2 Planeamento

8.2.1 Primeiro Semestre

Analisando o trabalho desenvolvido ao longo do primeiro semestre, o objetivo principal passou por cimentar as bases para a implementação, fase a realizar na segunda metade do ano letivo. O primeiro passo focou-se na compreensão do problema e na leitura de toda a documentação já existente sobre a plataforma SOCIAL. De seguida, foi realizada alguma pesquisa sobre o estado da arte, onde se inclui a participação na conferência AAL Fórum, sobre envelhecimento e tecnologia. A partir do final de Outubro pretendeu-se fazer o levantamento dos requisitos, de forma a passar às questões relacionadas com a arquitetura. A partir daí sentiu-se a necessidade de realizar experiências com as tecnologias a utilizar no projeto final, bem como definir a arquitetura do sistema.

Este planeamento também previa a escrita contínua, ao longo do semestre, do relatório de estágio, bem como a preparação da apresentação intermédia.

Nas duas figuras seguintes é possível comparar a calendarização das tarefas planeadas para o primeiro semestre com aquilo que foi realmente realizado.

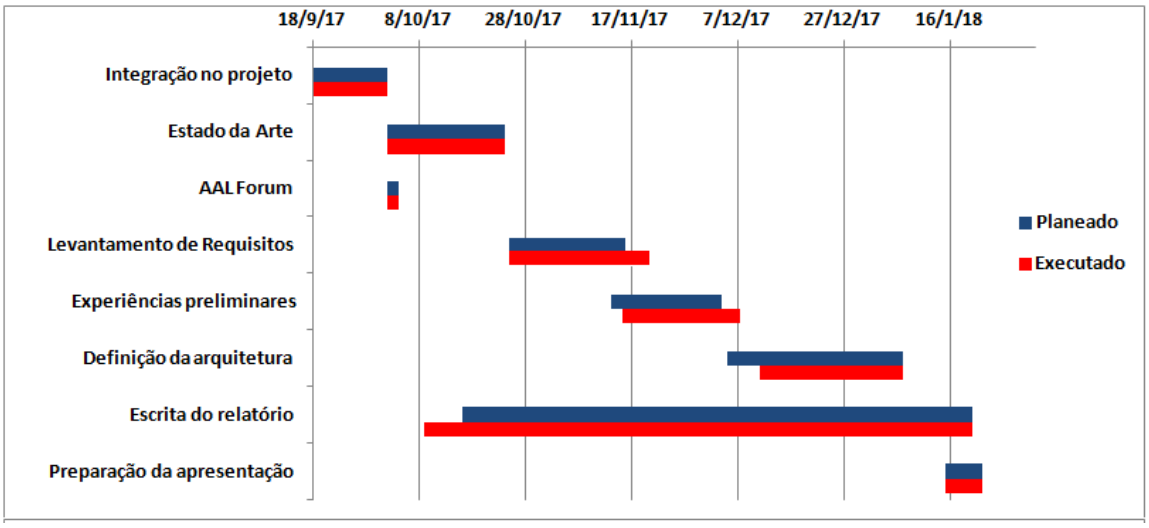

Figura 8.2 - Diagrama de Gantt das tarefas executadas no primeiro semestre.

De modo a perceber de que forma estava a ser distribuído o trabalho, também foi registado o esforço semanal, distribuído por seis categorias:

- **Requisitos** - Aqui está incluído o esforço relativo à integração no projeto, a pesquisa do Estado da Arte, a participação no AAL Forum e atividades relativas ao levantamento de requisitos.
- **Arquitetura** - Aqui estão incluídas as atividades relativas ao desenho e definição da arquitetura, às experiências sobre as tecnologias a utilizar e à análise tecnológica do projeto.
- **Implementação** - Aqui inclui-se todo o tempo que é despendido na implementação da solução final.
- **Testes** - Este ponto abrange todo o tempo gasto em testes ao projeto desenvolvido.
- **Gestão de Projeto** - Neste ponto estão incluídas atividades de gestão do projeto, como o planeamento, riscos, reuniões com o orientador do IPN-LIS sobre o trabalho a realizar e reuniões com o restante consórcio da plataforma SOCIAL.
- **Académico** - Corresponde ao esforço dedicado às tarefas relacionadas com a parte académica, como as reuniões com o orientador do DEI, a elaboração do relatório de tese ou a preparação da apresentação da mesma.

A Figura 8.3 representa o resultado da distribuição do trabalho semanal, por percentagem, ao longo do primeiro semestre, para cada uma das categorias mencionadas anteriormente. Ao contrário do diagrama de Gantt que mostra as diferentes etapas do projeto, com os intervalos de início e de fim de cada uma delas, este diagrama de esforço permite avaliar o trabalho realizado ao pormenor a cada semana. Além disso, este gráfico mostra o esforço

despendido em atividades não listadas no diagrama de Gantt, mas que são essenciais para o sucesso do projeto, como as reuniões com os orientadores. Como não foram realizadas atividades relacionadas com implementação e testes estas não constam neste gráfico.

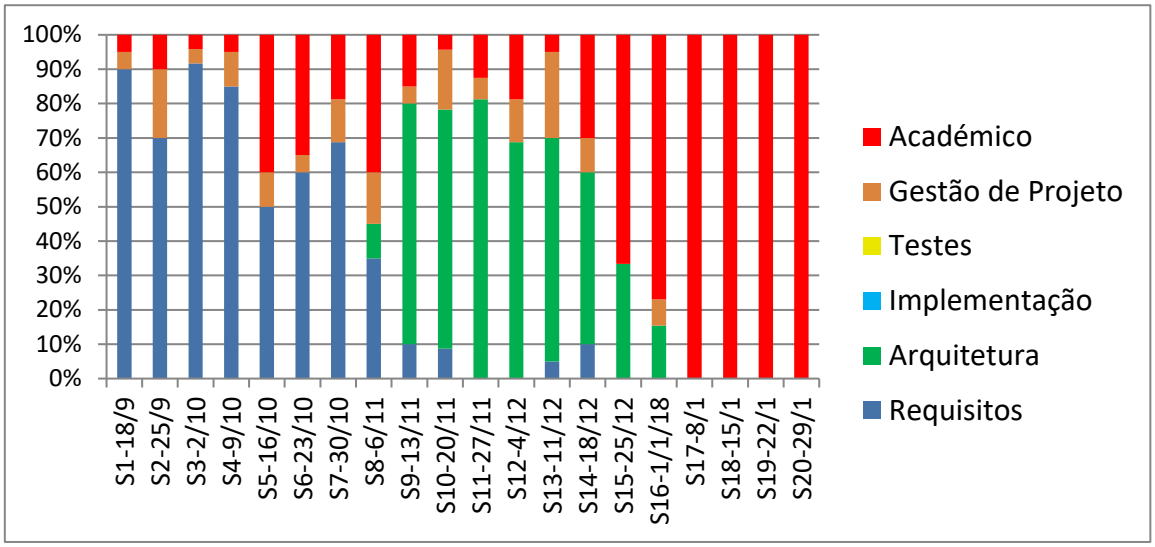

Figura 8.3 - Gráfico de distribuição do esforço semanal, por tarefa, no primeiro semestre.

8.2.2 Segundo semestre

De forma a planear o trabalho que ainda estava por realizar e minimizar os riscos relativos a atrasos e acumulação de tarefas, também foi elaborado um diagrama de Gantt para o segundo semestre.

O principal foco de trabalho incidia sobre a implementação e testes ao sistema, prevendo também a escrita do relatório final ao longo do semestre, bem como a preparação da apresentação na fase final do estágio. Este planeamento já incluía o aumento do horário do trabalho semanal, que vai passou das vinte para as quarenta horas semanais.

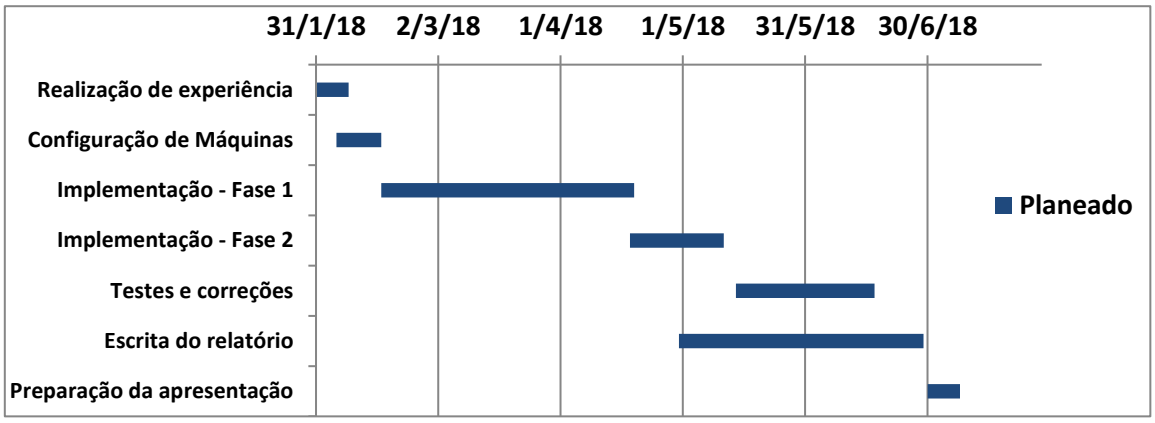

Figura 8.4 - Diagrama de Gantt das tarefas planeadas para o segundo semestre.

Após a apresentação intermédia foi necessário realizar algumas correções e alterações nos requisitos e na arquitetura do sistema. A necessidade de escrever um artigo até meados de Maio, também obrigou a alterações no planeamento, principalmente ao nível das fases de implementação e de testes. Foi necessário dividir a fase de testes e antecipar a primeira fase para testar as componentes desenvolvidas até à escrita do artigo. A Figura 8.5 mostra as alterações realizadas ao planeamento das tarefas para o segundo semestre e faz uma comparação entre o trabalho planeado e o que foi executado.

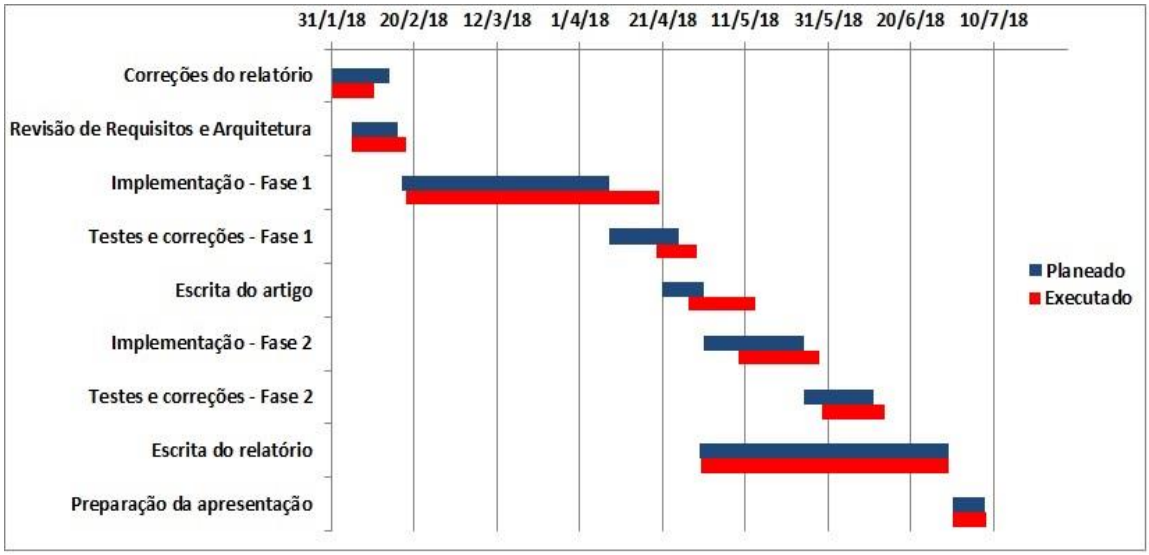

Figura 8.5 - Diagrama de Gantt das tarefas executadas no segundo semestre.

A partir da figura anterior é possível constatar que, apesar do atraso durante a primeira fase de implementação, não houve problemas para terminar o trabalho no período estipulado. Os atrasos existentes nas fases posteriores fizeram com que a última fase de testes terminasse um pouco depois do previsto, porém, a escrita do artigo, que coincidiu com o início da escrita do relatório, permitiu avançar mais do que o previsto, deixando menos trabalho para o final do semestre.

A Figura 8.6 apresenta a distribuição do esforço semanal por tarefa relativo ao semestre. Os dados observados coincidem com o diagrama de Gantt apresentado anteriormente, sendo possível detalhar o trabalho realizado ao longo do tempo.

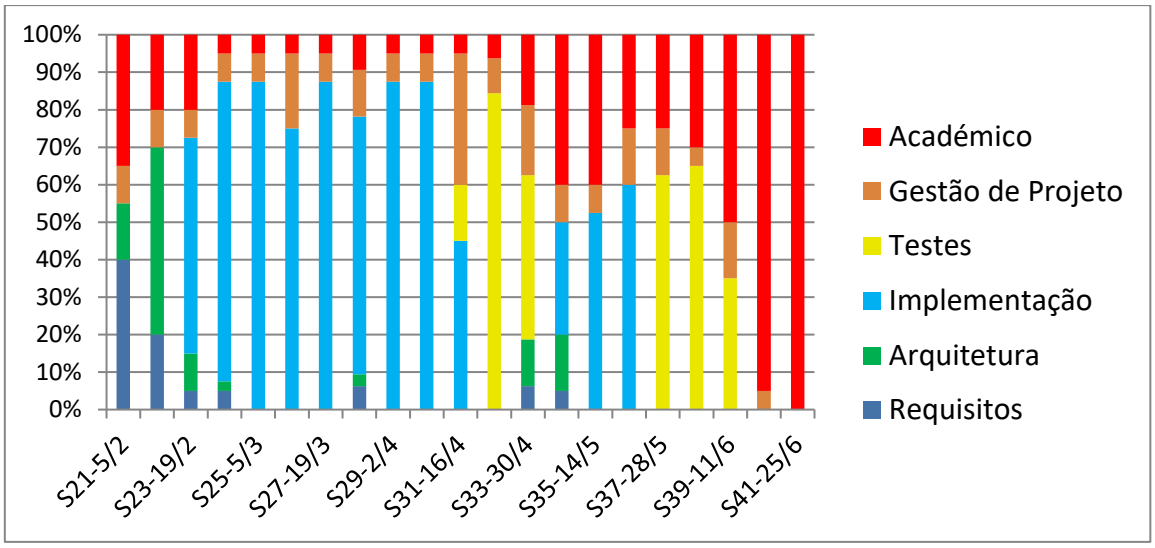

Figura 8.6 - Gráfico de distribuição do esforço semanal, por tarefa, no segundo semestre.

8.3 Riscos

Os riscos correspondem a eventos que podem ter um impacto negativo no objetivo do projeto. As suas fontes tanto podem estar diretamente relacionadas com o desenvolvimento de *software*, como com problemas com a equipa de desenvolvimento ou mesmo com a identificação deficiente dos riscos. Assim, é essencial que sejam identificados o mais cedo possível, para que seja possível responder a contrariedades e evitar que levem ao fracasso do projeto. Associados aos riscos existem dois fatores que permitem fazer uma melhor avaliação do perigo que representam para o trabalho a desenvolver:

- **Impacto** - Representa o grau de ameaça que terá no projeto e pode ser dividido nos seguintes níveis:
    - **Baixo** - Os objetivos do projeto não serão afetados de forma significativa.
    - **Médio** - Os objetivos do projeto podem ser atingidos, mas será necessário um esforço extra.
    - **Alto** - Os objetivos do projeto poderão estar comprometidos, levando ao fracasso do mesmo.
- **Probabilidade** - Representa o grau de possibilidade de um risco ocorrer e pode ser dividida nos seguintes níveis:
    - **Baixa** - A probabilidade de ocorrência é inferior a 25%.
    - **Média** - A probabilidade de ocorrência é superior a 25% e inferior a 75%.
    - **Alta** - A probabilidade de ocorrência é superior a 75%.

Mesmo após a mitigação de riscos de grande impacto durante a fase de experiências, ainda existem alguns que podem comprometer o sucesso do projeto. Na Tabela 8.1 é feita uma descrição de cada um deles, assim como o impacto e probabilidade de ocorrerem. De forma a prevenir a sua ocorrência, é também definido um plano de mitigação para cada um deles.

Tabela 8.1 - Tabela de identificação dos riscos do projeto.

ID	Descrição	Impacto	Probabilidade	Plano de mitigação	Estado
R1	Quantidade de dados insuficiente	Alto	Média	Caso exista algum problema com a recolha de dados a partir dos sensores, poderá ser usado o injetor de dados desenvolvido durante as experiências para simular o envio de dados.	Plano de mitigação executado
R2	Má estimação das tarefas	Médio	Média	Fazer revisões regulares do planeamento efetuado e fazer o controlo semanal do esforço despendido no projeto.	Resolvido
R3	Falta de comunicação com os restantes elementos do consórcio	Médio	Média	Marcação de reuniões mensais ou quinzenais para avaliar o estado atual do projeto.	Plano de mitigação executado
R4	Falta de conhecimento das tecnologias a utilizar	Alto	Baixa	Realização de tutoriais sobre as tecnologias que vão ser utilizadas.	Resolvido
R5	Número elevado de falsos positivos e falsos negativos	Alto	Média	Utilização da aplicação "*My Social*" para avaliar alertas e, mais tarde, treinar o algoritmo de tratamento de dados	Plano de mitigação não executado

Após uma análise aos riscos detetados, verificou-se que a grande maioria apresentava um impacto elevado, o que podia colocar em causa o sucesso do projeto. A quantidade insuficiente de dados foi um dos riscos que realmente aconteceu devido à falta de dados de sensores reais. Porém a execução do plano de mitigação permitiu que a construção da Componente de deteção de eventos não ficasse comprometida. A indefinição de alguns requisitos e da arquitetura obrigou à execução do plano de mitigação, que levou à realização de reuniões *task force* semanais para fechar esses pontos. A mitigação de falsos negativos e falsos positivos também era outra das preocupações para esta componente, mas com o plano de mitigação definido espera-se que esse risco não tenha impacto no futuro do projeto. Os restantes riscos definidos não se vieram a concretizar, acabando por ter um impacto nulo no resultado final.

Capítulo 9
Conclusão

Ao longo do último ano, foram aplicados, no contexto de um projeto real, muitos dos conhecimentos adquiridos no curso de Engenharia Informática. Após a integração no projeto SOCIAL e o estudo da área onde estava inserido, com a participação no AAL Forum e a pesquisa de projetos semelhantes, procedeu-se à definição dos requisitos e da arquitetura da componente a desenvolver. Estes processos ajudaram não só a compreender toda a estrutura do trabalho, mas também permitiram a aquisição de conhecimentos para mais tarde serem aplicados.

Os resultados obtidos após os testes realizados ao sistema de deteção de eventos significativos mostram que, para dados de sensores com uma percentagem igual ou inferior a 20% de ruído a cada dez minutos, todos os eventos previstos são detetados. Isso significa que a forma como está a ser tratado o ruído está a funcionar corretamente, indo ao encontro do resultado pretendido.

Os resultados mostraram também que, para a quantidade de dados referida no atributo relativo à velocidade de processamento dos dados, o sistema desenvolvido é capaz de processar os dados dentro dos valores referidos. Tendo em conta que os testes foram realizados localmente, na máquina onde foi implementada esta solução, com outros processos a executar em paralelo, dá ainda maior relevância aos resultados obtidos. Quando esta componente for integrada na plataforma SOCIAL, com um servidor dedicado, espera-se que seja capaz de responder ainda melhor para o mesmo número de dados recebidos.

Para trabalho futuro pretende-se implementar os casos de uso que na fase de elicitação de requisitos apresentavam uma prioridade mais baixa e que não tinham o seu desenvolvimento planeado para este estágio. Isto implicará implementar novos filtros, com recurso a novos sensores, como o consumo de eletricidade, consumo de água e *tag* de presença e testar as novas atividades ainda em ambiente laboratorial.

Ainda em Novembro deste ano, esta componente será testada em ambiente real nas habitações de alguns idosos, onde já serão utilizados os sensores desenvolvidos pela empresa Micro I/O. Com a persistência dos dados a ser também testada, já serão enviados os eventos significativos para os cuidadores, que poderão fazer uma avaliação desses alertas. Dessa

forma, será possível corrigir alguns falsos positivos e treinar o algoritmo antes do término do projeto.

No geral, todo o trabalho foi executado com sucesso, apesar de alguma indefinição na parte inicial do projeto. As alterações realizadas no início do segundo semestre ao nível do planeamento e implementação deram uma ideia mais concreta do trabalho final que se queria apresentar. Os planos de mitigação definidos para os riscos mostraram-se bastante importantes, porque alguns tiveram que ser executados e isso permitiu que esses problemas não tivessem um impacto tão severo no trabalho realizado.

A publicação de um artigo sobre o trabalho realizado na conferência HCist 2018 (*International Conference on Health and Social Care Information Systems and Technologies*), presente em anexo, é mais um sinal do enorme potencial de todo o trabalho que está a ser realizado. Tendo em conta as preocupações, muitas vezes associadas ao envelhecimento e ao crescimento desta faixa etária, esta solução poderá ajudar as associações de assistência social a perceber em que condições se encontra a população local. Com o projeto SOCIAL pretende-se que os cuidadores mantenham-se informados e que os familiares possam também ficar mais descansados.

Referências

[1] Instituto Nacional de Estatística (INE), «Mantém-se o agravamento do envelhecimento demográfico, em Portugal, que só tenderá a estabilizar daqui a cerca de 40 anos», *Inst. Nac. Estatística*, n. 2015, pp. 1–19, 2017.

[2] F. J. García-Peñalvo, M. Á. Conde, e V. Matellán-Olivera, «Mobile Apps for Older Users – The Development of a Mobile Apps Repository for Older People.», *Zaphiris P., Ioannou A. Learn. Collab. Technol. Technol. Environ. Learn. Collab. LCT 2014. Lect. Notes Comput. Sci. vol 8524. Springer, Cham*, pp. 117–126, 2014.

[3] J. Synnott, C. Nugent, e P. Jeffers, «Simulation of smart home activity datasets», *Sensors (Switzerland)*, vol. 15, n. 6, pp. 14162–14179, 2015.

[4] E. M. Tapia e S. S. Intille, «Activity Recognition in the Home Using Simple and Ubiquitous Sensors».

[5] P. Skocir, P. Krivic, M. Tomeljak, M. Kusek, e G. Jezic, «Activity Detection in Smart Home Environment», *Procedia Comput. Sci.*, vol. 96, pp. 672–681, 2016.

[6] L. Köping, K. Shirahama, e M. Grzegorzek, «A general framework for sensor-based human activity recognition», *Comput. Biol. Med.*, vol. 95, n. April 2017, pp. 248–260, 2018.

[7] A. G. Wilde, «An Overview of Human Activity Detection Technologies for Pervasive Systems».

[8] G. Sebestyen, I. Stoica, e A. Hangan, «Human activity recognition and monitoring for elderly people», *2016 IEEE 12th Int. Conf. Intell. Comput. Commun. Process.*, pp. 341–347, 2016.

[9] A. Moraru, M. Pesko, M. Porcius, C. Fortuna, e D. Mladenic, «Using Machine Learning on Sensor Data», pp. 341–347, 2010.

[10] M. Brodie, «Data quality in information systems», *Inf. Manag.*, vol. 3, pp. 245–258, 1980.

[11] A. Karkouch, H. Mousannif, H. Al Moatassime, e T. Noel, «Data quality in internet of things: A state-of-the-art survey», *J. Netw. Comput. Appl.*, vol. 73, pp. 57–81, 2016.

[12] M. Kara, O. Lamouchi, e A. Ramdane-Cherif, «A Quality Model for the Evaluation AAL Systems», *Procedia Comput. Sci.*, vol. 113, pp. 392–399, 2017.

[13] A. Loureiro, L. Torgo, e C. Soares, «Outlier Detection Using Clustering Methods : a data cleaning application», *Data Min. Bus. Work. Porto, Port.*, pp. 1–12, 2005.

[14] X. Hong e C. D. Nugent, «Segmenting sensor data for activity monitoring in smart environments», *Pers. Ubiquitous Comput.*, vol. 17, n. 3, pp. 545–559, 2013.

[15] «inCASA Project». [Em linha]. Disponível em: http://www.incasa-project.eu. [Acedido: 11-Out-2017].

[16] inCASA, «D6.6 Pilot Evaluation and inCASA Platform Validation and Recommendation», n. 250505, 2012.

[17] «Fit4work-About the project». [Em linha]. Disponível em: http://fit4work-aal.eu/abouttheproject.html. [Acedido: 11-Out-2017].

[18] M. Gjoreski e B. Cvetković, «Real-time Physical Activity and Mental Stress Management with a Wristband and a Smartphone», pp. 225–228, 2017.

[19] «Relaxed Care Project». [Em linha]. Disponível em: http://www.relaxedcare.eu. [Acedido: 12-Out-2017].

[20] «AAL Europe-CAMI Project». [Em linha]. Disponível em: http://www.aal-europe.eu/projects/cami/. [Acedido: 13-Out-2017].

[21] «CAMI Project». [Em linha]. Disponível em: http://www.camiproject.eu/. [Acedido: 13-Out-2017].

[22] «AAL Europe - eCAALYX». [Em linha]. Disponível em: http://www.aal-europe.eu/projects/ecaalyx/. [Acedido: 17-Out-2017].

[23] F. Nunes, M. Kerwin, e P. A. Silva, «Design Recommendations for TV User Interfaces for Older Adults: Findings from the eCAALYX Project», *Proc. 14th Int. ACM SIGACCESS Conf. Comput. Access. - ASSETS '12*, n. October, p. 41, 2012.

[24] «AAL Europe - Health@Home». [Em linha]. Disponível em: http://www.aal-europe.eu/projects/healthhome/. [Acedido: 18-Out-2017].

[25] I. Sanchez-Tato *et al.*, «Health @ Home: A telecare system for patients with chronic heart failure», *2010 Fifth Int. Conf. Broadband Biomed. Commun.*, pp. 1–5, 2010.

[26] A. Queirós *et al.*, «ICF inspired personas to improve development for usability and accessibility in Ambient Assisted Living», *Procedia Comput. Sci.*, vol. 27, pp. 409–418, 2013.

[27] A. I. Martins, A. Queirós, N. P. Rocha, T. Neves, A. Damasceno, e L. Arieira, «Personas and Scenarios to Improve the Development of an Electronic Social Record Platform», 2018.

[28] Archive.cnx.org, «Atributos de qualidade». [Em linha]. Disponível em: https://archive.cnx.org/contents/67b8bc5e-7e5e-4885-aee4-e57105b00f73@5/atributos-de-qualidade. [Acedido: 08-Jan-2018].

[29] «Apache Spark - RDD». [Em linha]. Disponível em: https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm. [Acedido: 09-Jan-2018].

[30] «FHIR Web Page». [Em linha]. Disponível em: http://hl7.org/fhir/modules.html. [Acedido: 08-Jan-2018].

[31] «Apache Spark Web Page». [Em linha]. Disponível em: https://spark.apache.org/. [Acedido: 09-Jan-2018].

[32] T. Das, M. Zaharia, e P. Wendell, «Diving into Apache Spark Streaming's Execution Model», *Engeneering Blog*, 2015. [Em linha]. Disponível em: https://databricks.com/blog/2015/07/30/diving-into-apache-spark-streamings-execution-model.html. [Acedido: 09-Jan-2018].

[33] «MQTT Web Site». [Em linha]. Disponível em: http://mqtt.org/. [Acedido: 09-Jan-2018].

[34] A. R. Earls, «XMPP: IoT protocol winner, or second place to MQTT?» [Em linha]. Disponível em: http://internetofthingsagenda.techtarget.com/feature/XMPP-IoT-protocol-winner-or-second-place-to-MQTT. [Acedido: 09-Jan-2018].

[35] «NoSQL: O que é, vantagens e limitações». [Em linha]. Disponível em: https://sirobotica.wordpress.com/2014/10/23/nosql-o-que-e-vantagens-e-limitacoes/. [Acedido: 09-Jan-2018].

[36] «XACML Documentation». [Em linha]. Disponível em: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html. [Acedido: 21-Jun-2018].

[37] A. J. Lattanze, *Architecting Software Intensive Systems: A Practitioners Guide*. Auerbach Publications, 2009.

[38] G. Fairbanks, «Just Enough Software Architcture: A Risk-Driven Approach», Marshal and Brainerd, 2010.

[39] N. Rani, «Review on Application of Pipe and Filter Architectural Style», *Int. J. Curr. Trends Eng. Res.*, vol. 2, n. 9, pp. 7–12, 2017.

[40] «LTSA Documentation». [Em linha]. Disponível em: https://www.doc.ic.ac.uk/ltsa/. [Acedido: 20-Jun-2018].

[41] A. B. Sharma, L. Golubchik, e R. Govindan, «Sensor Faults : Detection Methods and Prevalence in Real-World Datasets», *ACM Trans. Sens. Networks*, vol. 6, n. 3, p. 23, 2010.

Anexos

Anexo A

Artigo sobre a componente desenvolvida

CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2018

Testing a significant event detector in lab environment

João Rainho[ab], António Damasceno[a], Alberto Cardoso[b]

[a]*Instituto Pedro Nunes – Laboratory for Informatics and Systems, Rua Pedro Nunes, 3030-199 Coimbra, Portugal*

[b]*Department of Informatics Engeneering - University of Coimbra, Pinhal de Marrocos, 3030-229 Coimbra, Portugal*

Abstract

Nowadays, the lack of resources to provide social support for the elderly is a challenge of many societies. This work describes the development and testing of a significant event detection component to be used to process data from IoT sensors. The solution will be installed in the homes of the elderly, to detect events that are not normal in their daily routine. To test this significant event detector in a laboratory environment it was necessary to build a simulator capable of generating sensor data and compare with expected configurations. The results obtained show that the simulator injects significant abnormalities and left good expectations for the work to be carried out in the future, specifically the use of machine learning tools to further discover what is normal for each individual and raise more accurate notifications.

© 2018 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Selection and peer-review under responsibility of the scientific committee of the CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies.

Keywords: Internet Of Things; Data Injection; Data Processing; Events; Sensors; Simulator

1. Introduction

The use of sensors is important to obtain data about a particular person or environment. However, before an event is reached, it is necessary to analyze a large amount of data and check if the information obtained corresponds to reality. One of the examples where the use of sensors can be useful is in the monitoring of the elderly. This age group needs a frequent accompaniment, but resources are limited and often not enough to address all situations.

Several techniques for detecting events have been researched and developed. *Skocir et al* [1] use data from two simple sensors to monitor the presence of a person at home and monitor whether the door is open or closed. This installation is non-intrusive and is simple because the sensor node to which sensors are connected is battery powered. They proposed to approaches for activity detection: one based on artificial neural network and another based on a sliding window, similar to what will be used in our project.

In another project, *Köping et al* [2] developed a general framework for sensor-based human activity recognition. Using a smartphone to gather and transfer data from different sensors and a codebook-based feature learning approach it is possible to recognize different activities of different types of sensors.

In the context of a social care project and in order to solve some of the problems related to the lack of accompaniment of the elderly people, it was developed a Significant Event Detector (SED) to process and analyse sensors data sent by multiple IoT Gateways installed in elderly's houses. This tool will help social institutions to monitor their end-users. It will be capable of detecting anomalies in terms of end-user behavior and trigger notifications such as: the user is normally awake at this time of the day but we have no activity today, the user left home and did not return, the user left the bed and did not return.

To test all developed components in laboratory environment (Technology Readiness Level 4), it was necessary generate data similar to those that will be collected in real houses. To solve the lack of data, it was built a simulator capable of injecting data for different types of sensors. The simulator is configured with multiple house setups in terms of sensors types and sensor locations, multiple end-user expected behaviors and generates data simulations a specific number of days. Additionally, it is possible to have settings for different days and days of the week, insert some randomness based on temporal windows and insert abnormal events for a person on specific days. This last point is extremely important in order to validate the significant event detection tool.

To give even more realism to the simulation, it is possible to add noise to the generated sensor data. Some filters have been developed to help in processing and cleaning up the received data and at the end of the whole process compare if the data generated by the simulator corresponds to the initially predicted.

In this article we intend to show the work done in the construction of this simulator, the results, conclusions and future work that is still intended to be developed within this social care project.

2. Architecture

The construction of SED follow a Pipe and Filter architecture. This architectural style is composed by components, known as filters, working all in parallel [3]. They will transform or filter data, before passing it on using connectors known as pipes to other components of the software architecture. Beyond the concurrent execution, other advantages of this architecture are the simplicity and the ability to reuse the filters.

The system can be divided into three blocks, as shown in Figure 1. The first corresponds to the Data Source, which transmits the sensor data and feeds the Filters block, through a Data Stream communication channel. The second block contains all the filters necessary to handle the data. The operation of this block depends on the previous configuration of an xml file that will describe each of the filters used and the type of data that they allow to treat. Figure 2 shows an example configuration where each filter needs an id, an in and out routing key and the class corresponding to the data it will receive and treat. After determining person events, the data is passed to the Data Sinker block, where the data is already processed.

```
<filters>
<filter id="pbcam1" routingkeyin="house1.cam1.raw"  routingkeyout="house1.cam1.f1"      classname="SensorFilter"/>
<filter id="pbcam2" routingkeyin="house1.jan1.raw"  routingkeyout="house1.jan.f1"       classname="SensorFilter"/>

<filter id="h1clock" routingkeyin="house1.cam1.raw" routingkeyout="house1.check.state"  classname="PersonCheck"/>

<filter id="stcam1" routingkeyin="house1.cam1.f1"   routingkeyout="house1.cam1.state"   classname="SensorState"/>
<filter id="stjan1" routingkeyin="house1.jan1.f1"   routingkeyout="house1.jan1.state"   classname="SensorState"/>
```

Fig. 1 – High level architecture of the system

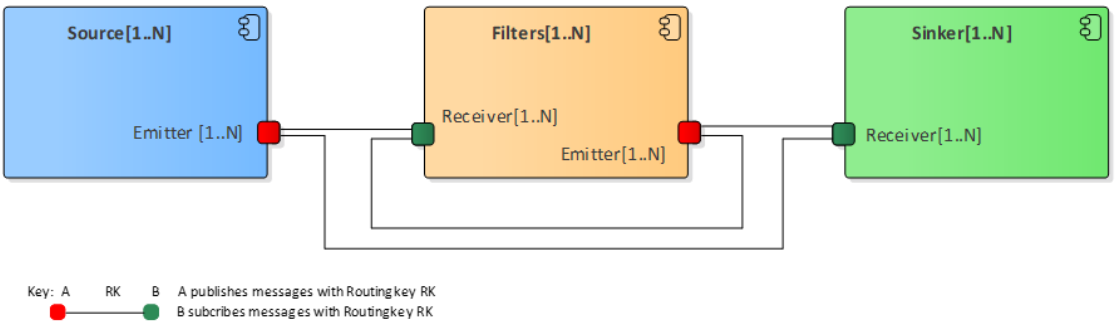

Fig. 2 – Example of .xml configuration filters

While simulation is running, SED starts to process the information. All generated data is sent to a topic, which allows messages to be sent to more than one subscriber. In the sending process, each message carries information about the sensor identifier, its status and the date on which this information was collected. In addition, a routing key is added, which will later help to identify and group sensor information and facilitating data processing. This key have the following format "House.SensorID.ProcessPhase".

For each of the processing phases, threads are created for each sensor in the simulation multiplied by the number of processing phases. This will allow that each open channel will be available to handle the data exclusively at any time, without creating confusion between the data.

The data of a particular sensor will be identified by its key and consumed to evaluate the state of the sensor. According to a study [4] there are four types of methods for identifying failures: rule-based methods, estimation methods, learning-based methods and time series analysis based methods. The last one enumerated was used to handle received data. Based on the last five events received from a sensor, the current value will be calculated by a voter and the result of majority will represent the actual state of the sensor at a given time. In this way, some false positives, caused by momentary changes in the state of the sensors, are avoided. For example, if a user is in bed but pressure sensor detect one state equal to 0, it will not be indicated immediately that person is not in bed. After being identified the value of the sensor in that moment will be sent to the topic, but with a different key, relative to the processing phase in which it is.

In the next phase of processing, the consumption of data from the topic is done in a similar way. However, based on previously obtained values, sensor events can be identified and the sensor status can be changed definitively. Thus, it is possible to compare the events configured with those that the simulation generated, and see if there are no errors in the process. This information will be added to the topic, with a new key, to be used next.

Finally, with the previous information, it will be possible to detect the events of the person, such as if a person is in bed or out of bed. This information will be sent one more time to the topic, but this time with the key in the format appropriate to an event of the person (PersonID.Sensor.State). In this way, the information can be used to learn the routines of a user and identify significant events, it is still possible to be consumed by more than one subscriber and to use it to generate more than one type of alert.

Figure 3 shows in detail, through a state machine, a filter to detect if a person is resting or not. Starting the system, the initial state is "Unknown", because no data has been received yet and it impossible to known the current state of the person. After receiving the first data from bed sensor, filter switch to the INBED state if user is in bed or NOTINBED, otherwise. If a person go to the bed, the time at that is recorded and it will be compared with the normal time to rest. If it is within normal time to resting, then the filter changes to the "Resting" state and will remain while the user is lying down. If the person rises, the next state will be "Resting NotInBed". If this situation occurs after the expected time to wake up, it will go to the "NOTINBED" state, otherwise it will remain until the expected time to wake up or return to the "Resting" state if a person return to bed. If the "Resting NotInBed" state last more than expected, an alert should be issued to the caregiver.

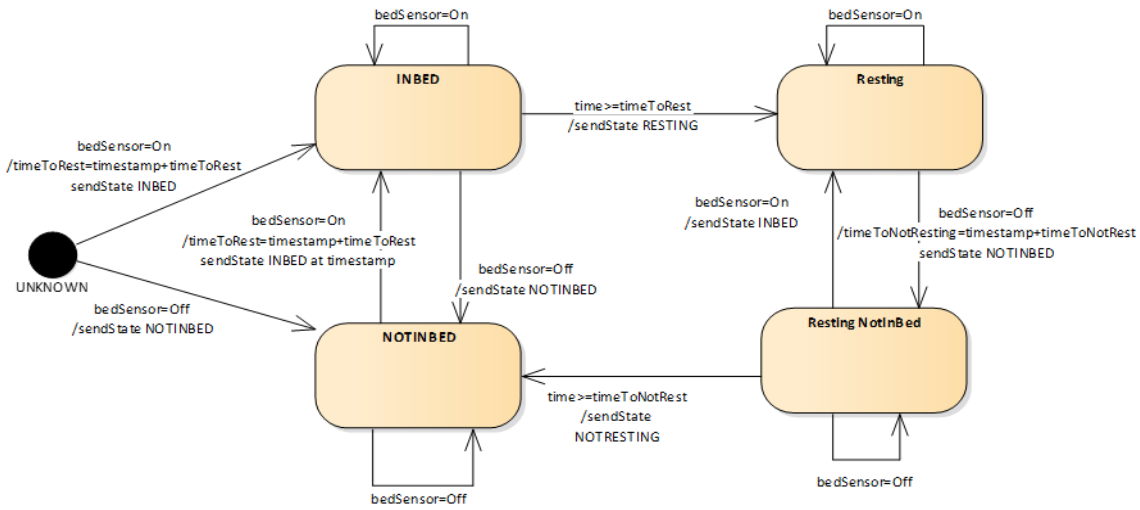

Fig. 3. State machine of Resting State Filter

3. Experimental Setup

The first phase of simulation will consist in the generation of data, based on the reading of an *.xml* file with information about the sensors in a given house and the habits of the people who inhabit it. In this file, it is also possible to set the simulation time, the frequency of data transmission and enter settings for different days, which leave the normal routine of the elderly and allow later send notification of a significant event. From this data it will be possible to generate people's events, which correspond to the changes in the state of a person (getting out of bed, lying in bed, etc.). Consequently, sensor's events can be generated, corresponding to the state changes of one or more sensors corresponding to the person's event (pressure sensor in bed or contact sensor in window changes state to 0 or 1). With all the events detected, it is possible generate all data between events.

```
<simulation start="2018-04-01 00:00:00" end="2018-04-02 23:59:00" increment="60000" pace="1">
    <house id="house1">
        <sensors>
            <sensor type="press" local="1" id="cam1" name="cama">
                <anomalies>
                    <anomaly type="switch" pattern=".*:.5:.*"/>
                </anomalies>
            </sensor>
            <sensor type="contact" local="1" id="jan1" name="janela quarto"></sensor>
        </sensors>
        <persons>
            <person name="Maria Trincha" id="person1" mintimesleep="20:00">
                <personconfigs levels="2">
                    <personconfig           type= "resting" base="1"  time="00:01"></personcor
                    <personconfig           type= "inbed" base="1"  time="00:01"></personconf:
                    <personconfig           type= "outofbed" base="1"  time="08:00"></personc
                    <personconfig           type= "notresting" base="1"  time="08:02"></persor
                    <personconfig level="1" type= "outofbed" base="1"  time="03:00"></personc
                    <personconfig level="1" type= "inbed" base="1"  time="03:10"></personconf:
                    <personconfig level="2" type= "outofbed" base="1"  time="05:00"></personc
                    <personconfig level="2" type= "inbed" base="1"  time="05:10"></personconf:
                </personconfigs>
            </person>
        </persons>

    </house>
```

Fig. 4. Example of simulation configuration

There are three different modes of simulation: a default mode, where user define the user's routines for a given task and it is equal for every day, another where it is possible specify the tasks by days and days of the week and, finally, a random mode, where user can define a range at which an event can occur. This will allow a certain event not to always occur at the same time and thus give greater variability to the data. However, the data received by real sensors may, sometimes, have some noise. In addition to being able to generate data without any noise, it is also possible to introduce noise or generate random noise, making the simulation even more real.

Figure 5 shows how the whole system works sequentially to detect if an elderly is resting or not resting. In order to detect this events it is necessary data from one sensor in bed and another in window. The green components correspond to the simulator and, in orange, are represented the filters that make up the SED.

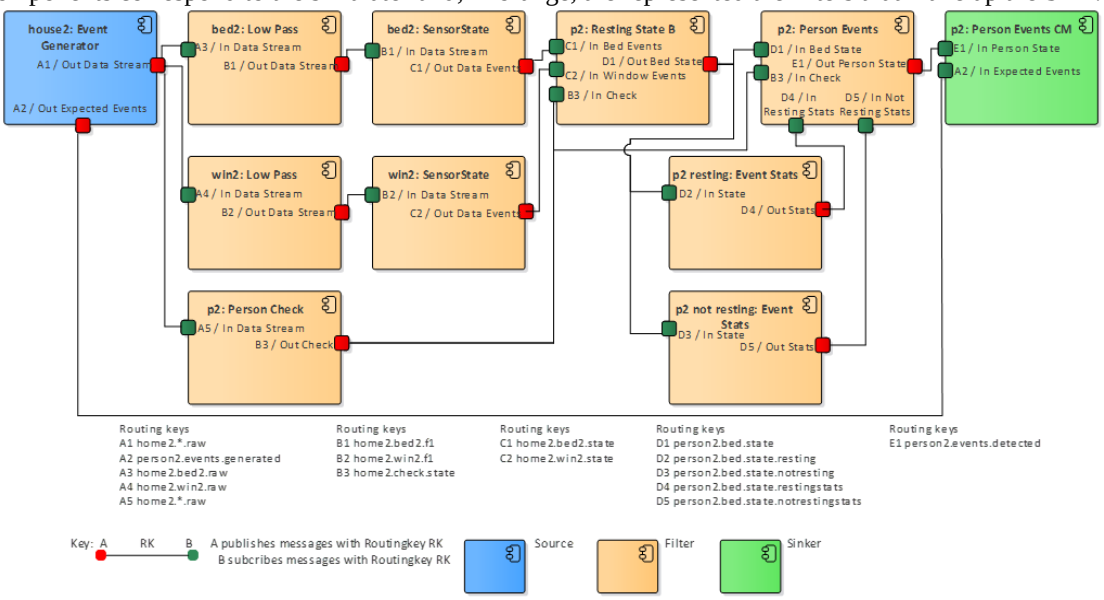

Fig. 5. Sequential plan to detect if an elderly is resting or not resting

As explained earlier in this chapter, filters have an essential role in the simulator architecture. The filters developed up to the time this article was written are presented below in resume:

- **Low Pass -** This filter will be responsible for selecting only the sensor data that is relevant to the detection of a particular person event.
- **Person Check** - This filter will be responsible for periodically checking the status of a person. This allow to update information about a person or generate alerts to caregivers.
- **Sensor State** – As previously described, this filter generates the correct current state of the sensor, based on the analysis of the last five values received.
- **Resting State** – Based on the current state of the person and the received values of the sensors at a given instant, this filter will detect if a person is resting or not resting.
- **Event Stats** – Based on the timestamp of each detected person event, total mean and standard deviations will be calculated as well as for each day of the week.
- **Person Events -** Based on the information sent by the PersonCheck, RestingState, and EventStats filters, it will be generated the person event detected and update the user's routine record.

In order to understand the precision of the developed filters, it was used confusion matrices, represented in Figure 1 as Person Events CM. This tables shows the performance of the algorithm and reports the number of false positives, false negatives, true negatives and true positives. Each row is composed by an expected event person, based on the pre-defined configuration, and each column corresponds to the same event, but resulting from the execution of the simulation. With this approach it will be possible to perceive the percentage of maximum noise in which it is possible to detect all the events, that is, without detecting any false positives and false negatives.

4. Results

After the experiments with the simulator and the significant event detector, it was possible to observe that the data generated by the simulator coincided completely with what was initially configured, without introducing noise into the received data.

The initial experiments were designed to validate that the system correctly identifies a reduced set of end-user behaviors (in bed, not in bed, resting, not resting) based on a single sensor data and the filter described above. The simulation was executed for a period of 10 days. In order to illustrate the results we chose to report 6 executions. One execution with no noise (A), five with noise generated, switching the expected on of state, at every minute ending in 5 (B), every minute ending in 3 (C), every minute ending in 3 or 7 (D), every minute ending in 3, 4, 5 or 7 (E), on a specific day F. Results are as follows;

Table 1. Results of simulation tests

Event	State	A	B	C	D	E	F
Resting	Detected	10	10	10	10	0	8
	Not detected	0	0	0	0	10	2
	Not expected	0	2	0	0	226	2
Not resting	Detected	10	10	10	10	1	9
	Not detected	0	0	0	0	9	1
	Not expected	0	2	0	0	225	1
In bed	Detected	30	30	30	30	21	26
	Not detected	0	0	0	0	9	4
	Not expected	0	0	0	0	1402	3
Not in bed	Detected	30	30	30	30	30	27
	Not detected	0	0	0	0	0	3
	Not expected	0	0	0	0	1392	2

As expected without noise the expected events were detected. With 10% noise (B, C, D) the system detected in one case four events not expected. The team is drilling down on the info to improve the system as these results were not expected. With 40% noise (E) the system is not reliable. And finally when the sensor does not work in a specific day (F) the results are the expected relative to the failure.

5. Conclusions and Future Work

In the future, the main target is to develop more filters and use other sensors like a contact sensor in the doors, a movement sensor or a sensor to measure electrical consumption in home appliances. Another category of filters will be added to allow the system to learn user behavior from sensor data. This will allow to detect even more events and make the final product as real as possible. Completed all tests in laboratory environment, the final step will be to install sensors in the homes of the elderly people and advance with Significant Event Detector testing in real environment.

Acknowledgements: To be added in the final paper.

References

[1] P. Skocir, P. Krivic, M. Tomeljak, M. Kusek, and G. Jezic, "Activity Detection in Smart Home Environment," *Procedia Comput. Sci.*, vol. 96, pp. 672–681, 2016.

[2] L. Köping, K. Shirahama, and M. Grzegorzek, "A general framework for sensor-based human activity recognition," *Comput. Biol. Med.*, vol. 95, no. April 2017, pp. 248–260, 2018.

[3] N. Rani, "Review on Application of Pipe and Filter Architectural Style," *Int. J. Curr. Trends Eng. Res.*, vol. 2, no. 9, pp. 7–12, 2017.

[4] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor Faults : Detection Methods and Prevalence in Real-World Datasets," *ACM Trans. Sens. Networks*, vol. 6, no. 3, p. 23, 2010.

Anexo B

Resultados dos testes às máquinas de estados

- **TM1 - Teste à Máquina de estados para detetar se um utilizador está a descansar (sensor de cama)**

```
Composition:
DEFAULT = UNKNOWN
State Space:
 5 = 2 ** 3
Analysing...
Depth 4 -- States: 5 Transitions: 11 Memory used: 48875K
No deadlocks/errors
Analysed in: 2ms
```

Figura B1 – *Safety test* TM1

```
Composition:
DEFAULT = UNKNOWN
State Space:
 5 = 2 ** 3
Progress Check...
-- States: 5 Transitions: 11 Memory used: 50703K
No progress violations detected.
Progress Check in: 1ms
```

Figura B2 – *Progress test* TM1

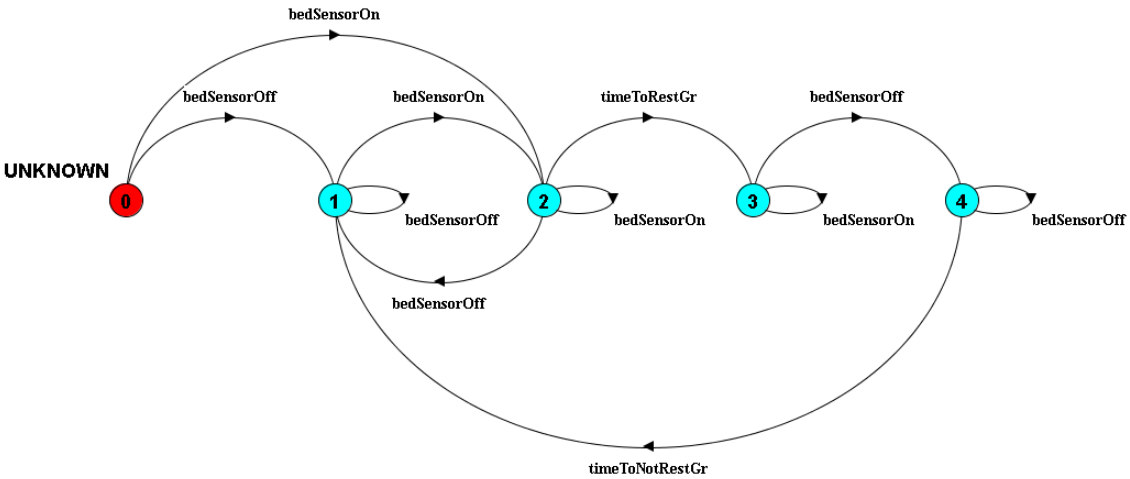

Figura B3 – Plano de execução do teste TM1

- **TM2 - Teste à Máquina de estados para alarme de cama**

```
Composition:
DEFAULT = BEDALARM
State Space:
 7 = 2 ** 3
Analysing...
Depth 3 -- States: 7 Transitions: 12 Memory used: 19313K
No deadlocks/errors
Analysed in: 1ms
```

Figura B4 – *Safety test* TM2

```
Composition:
DEFAULT = BEDALARM
State Space:
 7 = 2 ** 3
Progress Check...
-- States: 7 Transitions: 12 Memory used: 21925K
No progress violations detected.
Progress Check in: 1ms
```

Figura B5 – *Progress test* TM2

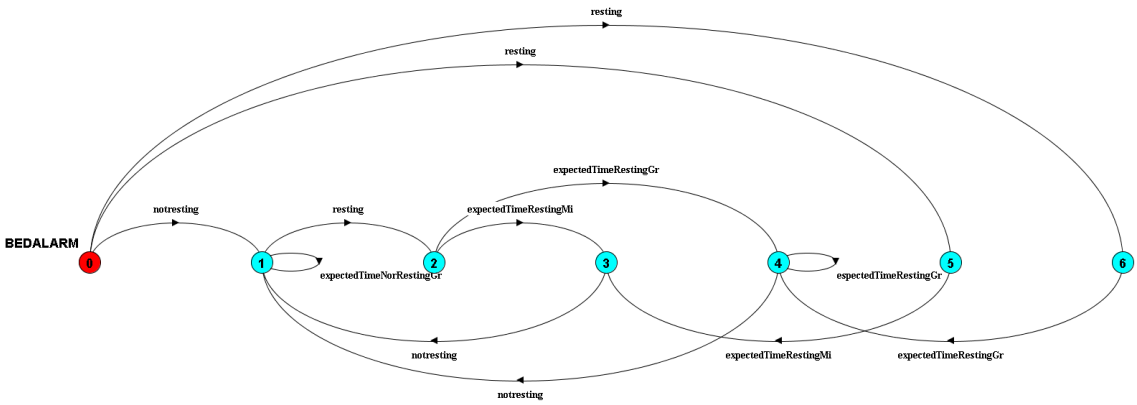

Figura B6 – Plano de execução do teste TM2

- **TM3 - Teste à Máquina de estados para detetar presença de utente em casa**

```
Composition:
DEFAULT = UNKNOWN
State Space:
 13 = 2 ** 4
Analysing...
Depth 4 -- States: 13 Transitions: 29 Memory used: 21608K
No deadlocks/errors
Analysed in: 4ms
```

Figura B7 – *Progress test* TM3

```
Composition:
DEFAULT = UNKNOWN
State Space:
 13 = 2 ** 4
Progress Check...
-- States: 13 Transitions: 29 Memory used: 18973K
No progress violations detected.
Progress Check in: 0ms
```

Figura B8 – *Progress test* TM3

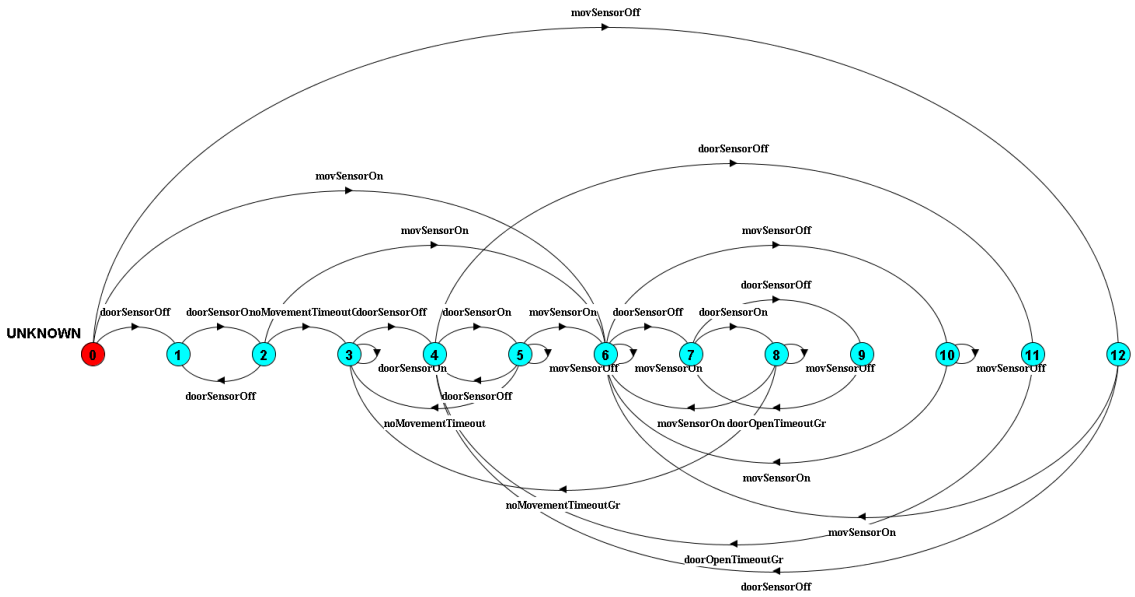

Figura B9 – Plano de execução do teste TM3

- **TM4 - Teste à Máquina de estados para alarmes relacionados com presença de utente em casa**

```
Compiled: DOORALARM
Composition:
DEFAULT = DOORALARM
State Space:
 3 = 2 ** 2
Analysing...
Depth 2 -- States: 3 Transitions: 6 Memory used: 19028K
No deadlocks/errors
Analysed in: 0ms
```

Figura B10 – *Progress test* TM4

```
Composition:
DEFAULT = DOORALARM
State Space:
 3 = 2 ** 2
Progress Check...
-- States: 3 Transitions: 6 Memory used: 28720K
No progress violations detected.
Progress Check in: 0ms
```

Figura B11 – *Progress test* TM4

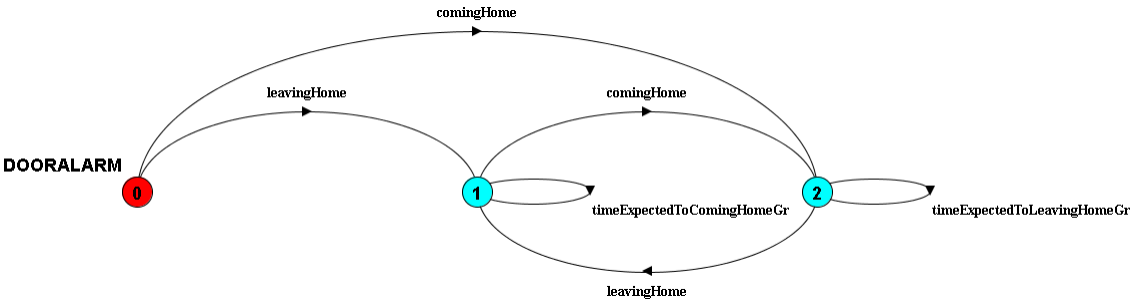

Figura B12 – Plano de execução do teste TM4

- **TM5 - Teste à Máquina de estados para detetar o estado da televisão**

```
Composition:
DEFAULT = PERSONTVSTATE
State Space:
 4 = 2 ** 2
Analysing...
Depth 2 -- States: 4 Transitions: 9 Memory used: 18875K
No deadlocks/errors
Analysed in: 1ms
```

Figura B13 – *Progress test* TM5

```
Composition:
DEFAULT = PERSONTVSTATE
State Space:
 4 = 2 ** 2
Progress Check...
-- States: 4 Transitions: 9 Memory used: 21225K
No progress violations detected.
Progress Check in: 1ms
```

Figura B14 – *Progress test* TM5

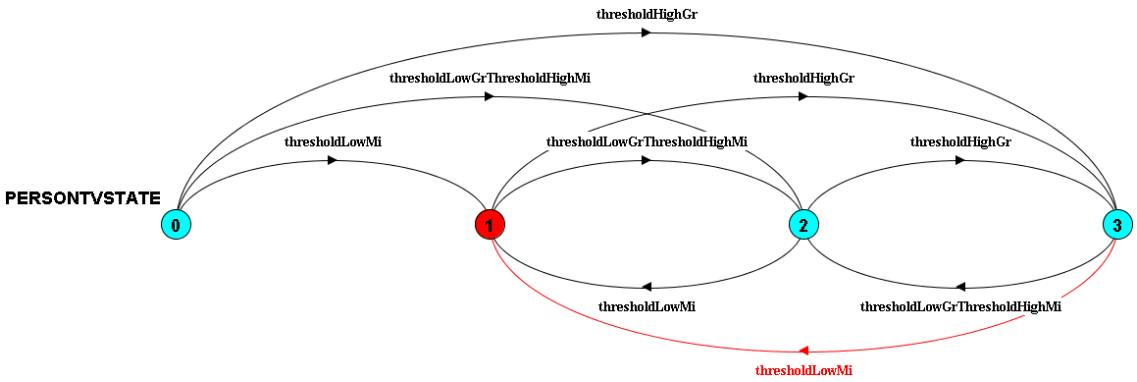

Figura B15 – Plano de execução do teste TM5

- **TM6 - Teste à Máquina de estados para aletas relacionados com a porta e a televisão**

```
Compiled: ALERTDOORTV
Composition:
DEFAULT = ALERTDOORTV
State Space:
 7 = 2 ** 3
Analysing...
Depth 3 -- States: 7 Transitions: 16 Memory used: 18537K
No deadlocks/errors
Analysed in: 0ms
```

Figura B16 – *Progress test* TM6

```
Composition:
DEFAULT = ALERTDOORTV
State Space:
 7 = 2 ** 3
Progress Check...
-- States: 7 Transitions: 16 Memory used: 21411K
No progress violations detected.
Progress Check in: 0ms
```

Figura B17 – *Progress test* TM6

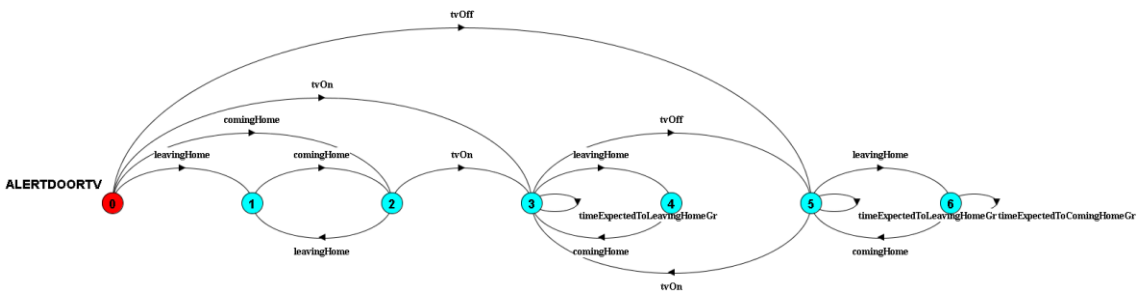

Figura B18 – Plano de execução do teste TM6

- **TM7 - Teste à Máquina de estados para detetar se o utente está a descansar (sensor na cama e na janela)**

```
Composition:
DEFAULT = UNKNOWN
State Space:
 5 = 2 ** 3
Analysing...
Depth 4 -- States: 5 Transitions: 15 Memory used: 18974K
No deadlocks/errors
Analysed in: 1ms
```

Figura B19 – *Progress test* TM7

```
Composition:
DEFAULT = UNKNOWN
State Space:
 5 = 2 ** 3
Progress Check...
-- States: 5 Transitions: 15 Memory used: 20690K
No progress violations detected.
Progress Check in: 0ms
```

Figura B20 – *Progress test* TM6

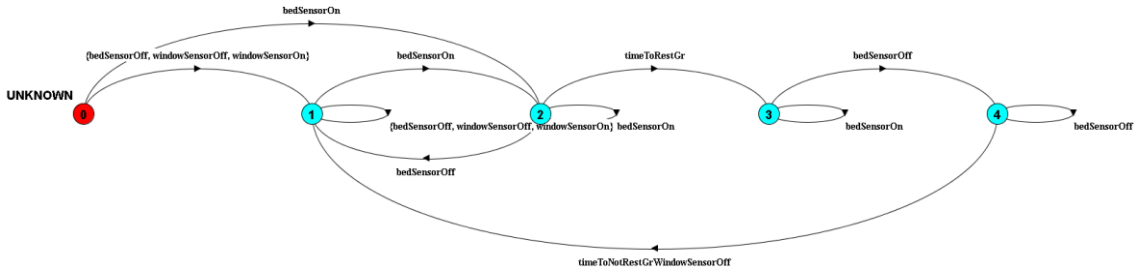

Figura B21 – Plano de execução do teste TM7