

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
Escola de Ciências e Tecnologia

**Adaptação dinâmica e sensível ao contexto de interfaces
móveis em ambiente ubíquo**

Tese de Doutoramento em Informática

André Filipe Esteves de Sousa

Orientador:

Professor Doutor Hugo Alexandre Paredes Guedes da Silva

Coorientador:

Professor Doutor João Manuel Pereira Barroso



Vila Real, 2017

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

**Adaptação dinâmica e sensível ao contexto de interfaces
móveis em ambiente ubíquo**

Tese de Doutoramento em Informática

André Filipe Esteves de Sousa

Orientador:

Professor Doutor Hugo Alexandre Paredes Guedes da Silva

Coorientador:

Professor Doutor João Manuel Pereira Barroso

Composição do Júri:

Doutor Manuel Pérez Cota, Professor Catedrático da *Escuela de Ingeniería Industrial da Universidad de Vigo*

Doutor José Paulo Barroso de Moura Oliveira, Professor Associado com Agregação da *Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro*

Doutor António Joaquim da Silva Teixeira, Professor Associado da *Universidade de Aveiro*

Doutor Paulo Nogueira Martins, Professor Auxiliar com Agregação da *Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro*

Doutor Hugo Alexandre Paredes Guedes da Silva, Professor Auxiliar com Agregação da *Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro*

Doutor António Manuel Nestor Ribeiro, Professor Auxiliar da *Escola de Engenharia da Universidade do Minho*

Doutor Vítor Manuel Pereira Duarte dos Santos, Professor Auxiliar Convidado do *Instituto Superior de Estatística e Gestão de Informação da Universidade Nova de Lisboa*.

Presidência do Júri:

Doutor Vitor Manuel de Jesus Filipe, Professor Associado com Agregação da Universidade de Trás-os-Montes e Alto Douro

Vila Real, 2017

O trabalho desenvolvido nesta tese foi financiado pela *Fundação para a Ciência e Tecnologia* na forma de uma bolsa de doutoramento com a referência **SFRH/BD/87259/2012**

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

RESUMO

A evolução da tecnologia veio permitir um aumento na qualidade de vida, na forma como interagimos entre nós e com o mundo. Esta evolução, permitiu-nos ultrapassar diversos desafios, mas, também criou novos obstáculos que não existiam anteriormente. No entanto, esta mesma tecnologia, que aumenta a eficiência das nossas atividades e nos disponibiliza informação à distância de um toque, também traz frustrações na sua utilização, em situações em que necessitamos de um conjunto específico de informação. A solução para estes problemas, passa pela deteção do contexto em que a aplicação está a ser usada, realizando as ações pretendidas, de forma otimizada e de acordo com esse contexto. Estas respostas passam por adaptações dinâmicas da interface, deduzindo as intenções do utilizador e adequado a interface, em particular, da forma como a informação é apresentada e de como são recebidos comandos do utilizador. Esta inferência do contexto de uso pode ser realizada pelo processamento de informação, a partir de diversas fontes, tais como: o perfil do utilizador, o planeamento a realizar e o histórico de uso.

Nesta tese é apresentada uma camada intermédia de software (sistema de middleware) multiplataforma, ADAPT, que tem como objetivo recolher e processar dados, inferir e facilitar a compreensão do contexto do utilizador e aplicar essas inferências às aplicações móveis. Os dispositivos móveis atuais possuem capacidades computacionais e sensores bastante avançados. No entanto, apresentam diversas limitações que devem ser tomadas em conta, tais como: limitações energéticas, comunicação, tipo de uso, entre outras. Devido a todas estas limitações, a inferência e aplicação de informação contextual nas aplicações é um processo complexo que é simplificado pela capacidade de abstração de sensores, algoritmos e processos disponibilizada pelo ADAPT. Assim, os programadores de aplicações móveis podem dispor

dos benefícios que a adaptação contextual proporciona para aumentar a eficiência e conforto do utilizador.

Nesta tese é proposto um modelo para a sensorização contextual do utilizador: o modelo contextual antecipatório, conceito que leva à proposta da arquitetura ADAPT. Esta pretende demonstrar e validar a utilização do ADAPT com um caso de estudo, desenvolvido para melhorar a aplicação de alunos da Universidade de Trás-os-Montes e Alto Douro, com os benefícios da adaptação contextual que permite realizar testes com utilizadores reais, em situações reais. A principal contribuição desta tese é o modelo teórico e arquitetural para a implementação e avaliação de um sistema de middleware para antecipação contextual.

Palavras Chave: Sensibilidade ao contexto, Adaptação contextual, Contexto, Inferência contextual, Middleware contextual, Aplicação móvel;

ABSTRACT

The evolution of technology has allowed an increase in quality of life, the way we interact among ourselves and with the world. This development has allowed us to solve many challenges, but also created new obstacles that did not previously exist. However, this same technology that improves efficiency and provides information at a distance of a touch, also bring frustrations in its use on different situations that we need a certain specific information. The solution to these problems is the detection of context, in which the application is being used and at the same time address the obstacles, so that it can allow the user intention to be the most efficient possible. One of the solutions to this problem is through dynamic adaptation of the interface and the way information is passed to the user. This inference the context of utilization can be achieved by processing different sources of information, such as user profile, physical sensors and usage history.

This thesis presents a multiplatform middleware system, ADAPT, which aims to collect data to facilitate the understanding of the user's context and apply those inferences to mobile applications. The current mobile devices have computing capabilities and highly advanced sensors, however, it presents a number of considerations that must be taken into account as energy constraints, communication support, type of use, among others. Because of all these restrictions, the inference and contextual awareness in mobile applications is a complex process that is simplified by sensors, algorithms and processes abstraction provided by ADAPT. So it is possible to allow mobile developers, to take advantage from the benefits that contextual adaptation may allow to increase efficiency and user comfort.

This thesis proposes a model for contextual user sensing: The contextual model anticipatory concept that leads to the proposal from ADAPT architecture. This thesis demonstrates the use of the ADAPT system with a case study designed to improve the application of students from the University of Trás-os-Montes and Alto Douro with the benefits of contextual adaptation to perform tests with real users in real situations. The main contribution of this thesis is the theoretical and architectural model for the implementation and evaluation of a middleware system for contextual anticipation.

Keywords: context-aware/awareness, contextual adaptation, context, contextual inference, contextual middleware, mobile application;

AGRADECIMENTOS

Durante a nossa vida vamos encontrando obstáculos e desafios que ultrapassamos, por vezes, graças a algumas palavras de apoio e incentivo que nos motivam a continuar. Por isso, gostaria de deixar aqui registado o meu agradecimento a todas as pessoas que, de uma maneira ou de outra, me deram o apoio necessário durante este desafio:

Agradeço à **Fundação para a Ciência e Tecnologia (FCT)**, por ter acreditado no meu projeto e me ter apoiado sob a forma de uma bolsa de doutoramento, que permitiu financiar os meus estudos durante esta fase.

Obrigado ao **Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência (INESC TEC)** e à **Universidade de Trás-os-Montes e Alto Douro (UTAD)**, por todo o apoio institucional, sob a forma de meios e serviços, que me disponibilizaram. Assim como aos seus colaboradores que, de alguma forma, tiveram um papel na minha formação.

Quero deixar um agradecimento especial aos meus orientadores, o **Professor Hugo Paredes**, e o **Professor João Barroso**, por todo o apoio e orientação, especialmente naqueles momentos de maior insegurança e em que nos sentimos perdidos no meio de centenas de ideias. O vosso apoio foi fundamental para descobrir qual o caminho e, sem vocês, não seria possível ter chegado até aqui.

Gostaria de deixar também um agradecimento aos **professores** da UTAD, que de uma maneira ou de outra, tiveram influência no meu percurso e no trabalho realizado, ao **Centro de Informática da UTAD**, em especial ao **SIDE** e, em particular, ao **Jorge Borges**, pelo apoio e acesso disponibilizado para o desenvolvimento do caso de estudo realizado.

Gostaria de deixar um agradecimento especial ao **Hugo Fernandes, Arsénio Reis, José Faria e a todos os amigos**, que apesar de não mencionados, me aturaram durante estes anos, (sei que não fui fácil).

Pretendo deixar o maior agradecimento possível aos **meus pais** e à **minha irmã**, pois sem o vosso apoio, esforço, dedicação e educação, nenhum dos meus sonhos poderia ser realizado, nem teria sido possível ter chegado aqui. Um bem-haja também, à minha restante família que acreditou em mim e, de uma forma ou outra, me apoiaram neste percurso.

Por fim, quero deixar escrito o maior agradecimento à pessoa que, pela sua presença, iluminou os dias mais escuros, me motivou a ser melhor que eu mesmo, a ultrapassar todos os meus limites e a acreditar que, se consigo sonhar, consigo tornar realidade. Ela é a pessoa que mesmo nos piores momentos, se manteve sempre a meu lado com uma palavra amiga e motivadora. Muito obrigado à minha namorada, amiga e parceira, **Mónica Pinheiro**, por todo o apoio, paciência, amor e compreensão. És a melhor parte da minha vida, obrigado por seres quem és e, principalmente, por existires na minha vida.

A todos os mencionados e aos que, por lapso não foram mencionados, um muito obrigado por tudo o que fizeram e representam na minha vida.

A todos,

MUITO OBRIGADO

André Sousa

*Dedico esta tese aos meus pais pois sem o seu árduo esforço e
dedicação nada disto seria possível.*

ÍNDICE

Resumo	vii
Abstract	ix
Agradecimentos.....	xi
Lista de figuras.....	xix
Lista de tabelas.....	xxi
Acrónimos	xxiii
1. Introdução.....	1
1.1. Enquadramento.....	3
1.2. Motivação.....	4
1.3. Tese.....	5
1.4. Objetivos e contribuição.....	6
1.5. Metodologia.....	8
1.6. Organização da tese.....	11
2. Conceitos de contexto.....	13
2.1. Enquadramento.....	15
2.2. Contexto.....	19
2.2.1. Introdução.....	19
2.2.2. Revisão das definições de contexto	21
2.2.3. Definição de contexto	25

2.2.4.	Considerações finais	27
2.3.	Sensibilidade ao contexto.....	28
2.3.1.	Introdução.....	28
2.3.2.	Sensibilidade ao contexto	28
2.3.3.	Considerações finais	30
2.4.	Ciclo de vida do contexto.....	31
2.4.1.	Introdução.....	31
2.4.2.	Aquisição de Contexto.....	32
2.4.3.	Modelação de contexto.....	34
2.4.4.	Inferência de contexto.....	39
2.4.5.	Considerações Finais	46
2.5.	Desafios das técnicas de inferência contextual	47
2.5.1.	Localização interna.....	47
2.5.2.	Reconhecimento de atividade humana	52
2.6.	Considerações finais do capítulo.....	57
3.	Estado da arte de middleware contextual	59
3.1.	Enquadramento.....	61
3.2.	CORTEX	65
3.3.	Context Studio	66
3.4.	ContextPhone	67
3.5.	AWARENESS.....	68
3.6.	MyExperience.....	69
3.7.	CenceMe	70
3.8.	SociableSense	71
3.9.	Empath.....	72
3.10.	Funf.....	73
3.11.	SystemSens	74
3.12.	OpenDataKit	75

3.13.	Ohmage.....	76
3.14.	AWARE.....	77
3.15.	Ginger.io.....	78
3.16.	Considerações finais do capítulo.....	79
4.	Modelo contextual antecipatório.....	83
4.1.	Enquadramento.....	85
4.2.	Desafios e diretrizes orientadoras.....	87
4.3.	Modelo de inferência contextual para um sistema antecipatório.....	90
4.4.	Influências no modelo contextual antecipatório.....	93
4.5.	Considerações finais do capítulo.....	95
5.	Arquitetura.....	97
5.1.	Enquadramento.....	99
5.2.	Vista global da arquitetura.....	100
5.3.	Vista funcional da arquitetura.....	102
5.3.1.	Gestão Contextual.....	105
5.3.2.	Canal de comunicação.....	113
5.3.3.	Fonte contextual.....	116
5.4.	Considerações finais do capítulo.....	134
6.	Caso de estudo e avaliação.....	135
6.1.	Enquadramento.....	137
6.1.1.	Domínio do problema.....	137
6.1.2.	Requisitos aplicativos.....	140
6.1.3.	Interação com a aplicação.....	142
6.2.	Implementação e desenvolvimento.....	145
6.2.1.	Processo de desenvolvimento e testes.....	146

6.2.2. Caso de estudo – AlunosUTAD.....	148
6.3. Validação do modelo	152
7. Conclusões e trabalho futuro.....	159
7.1. Conclusões.....	161
7.2. Linhas de orientação futura	163
8. Lista de publicações originais.....	165
9. Bibliografia	167

LISTA DE FIGURAS

FIGURA 1-1 – MODELO GERAL DAS FASES DE IMPLEMENTAÇÃO DA METODOLOGIA DE INVESTIGAÇÃO BASEADA EM DESIGN SCIENCE (ADAPTADO DE HEVNER & CHATTERJEE, 2010).....	9
FIGURA 2-1 – CICLO DE VIDA DO CONTEXTO.....	31
FIGURA 2-2 – TÉCNICAS DE LOCALIZAÇÃO INDOOR (ADAPTADO DEKJÆRGAARD, 2007).....	48
FIGURA 2-3 – LOCALIZAÇÃO BASEADA EM IMPRESSÃO DIGITAL POR WIFI (FIND PROJECT, 2016).....	49
FIGURA 2-4 – ILUSTRAÇÃO DE UMA ARQUITETURA MULTISENSORIAL UTILIZADA PARA O RECONHECIMENTO DE ATIVIDADE HUMANA (YANG ET AL., 2015)	53
FIGURA 3-1 – MODELO DE UTILIZAÇÃO DE CONTEXTO (DEY ET AL., 2001).....	62
FIGURA 3-2 – MODELO CONCEPTUAL DO SISTEMA CORTEX (BIEGEL & CAHILL, 2004).....	65
FIGURA 3-3 – MODELO CONCEPTUAL PARA CONSTRUÇÃO DE VOCABULÁRIOS DO CONTEXT STUDIO(KORPIPÄÄ ET AL., 2004)	66
FIGURA 3-4 – ARQUITETURA DO SISTEMA CONTEXTPHONE (RAENTO ET AL., 2005)	67
FIGURA 3-5 – ARQUITETURA DO SISTEMA AWARENESS (VAN SINDEREN ET AL., 2006).....	68
FIGURA 3-6 – ARQUITETURA CONCEPTUAL DO SISTEMA MYEXPERIENCE (FROEHLICH ET AL., 2007)	69
FIGURA 3-7 – ARQUITETURA CONCEPTUAL DO MODELO CENCEME (MILUZZO ET AL., 2008).....	70
FIGURA 3-8 – ARQUITETURA CONCEPTUAL DO SISTEMA SOCIABLESENSE (RACHURI ET AL., 2010).....	71
FIGURA 3-9 – ARQUITETURA CONCEPTUAL DO SISTEMA EMPATH (DICKERSON ET AL., 2011).....	72
FIGURA 3-10 – MODELO CONCEPTUAL DO SISTEMA FUNF (AHARONY ET AL., 2011)	73
FIGURA 3-11 – ARQUITETURA DO SISTEMA SYSTEMSENS (FALAKI ET AL., 2011).....	74
FIGURA 3-12 – ARQUITETURA CONCEPTUAL DO SISTEMA OPENDATAKIT (BRUNETTE ET AL., 2012).....	75
FIGURA 3-13 – ARQUITETURA CONCEPTUAL E MODULAR DO SISTEMA OHMAGE (TANGMUNARUNKIT ET AL., 2015)	76
FIGURA 3-14 – ARQUITETURA CONCEPTUAL DO SISTEMA AWARE (FERREIRA ET AL., 2015).....	77
FIGURA 3-15 – MODELO CONCEPTUAL DO PROJETO GINGER.IO (GINGER INC, 2016),	79
FIGURA 4-1 – ARQUITETURA CONCEPTUAL GENÉRICA DE UM MIDDLEWARE CONTEXTUAL	86
FIGURA 4-2 – FRAGMENTAÇÃO DE DISPOSITIVOS ANDROID REPARTIDO POR FABRICANTE (OPENSIGNAL, 2015).....	89
FIGURA 4-3 – ARQUITETURA CONCEPTUAL DO MODELO CONCEPTUAL ANTECIPATÓRIO.....	91
FIGURA 5-1 – MODELO CONCEPTUAL DE SUPORTE AO MODELO CONTEXTUAL ANTECIPATÓRIO	100
FIGURA 5-2 – VISTA GLOBAL DA ARQUITETURA DO ADAPT	101
FIGURA 5-3 – VISTA DE COMPONENTES DO ADAPT	104

FIGURA 5-4 – ESTRUTURA DA GESTÃO CONTEXTUAL	105
FIGURA 5-5 – VISTA DE SEQUÊNCIA DE PEDIDO DA INFORMAÇÃO CONTEXTUAL	106
FIGURA 5-6 – ESTRUTURA DE DADOS DO CATÁLOGO SENSORIAL	107
FIGURA 5-7 – ESTRUTURA DO MICROSERVIÇO GESTÃO SENSORIAL	108
FIGURA 5-8 – VISTA DE SEQUÊNCIA DO MODELO DE GESTÃO ENERGÉTICA.....	109
FIGURA 5-9 – PROCESSO INTERNO DE FUNCIONAMENTO DA FONTE CONTEXTUAL DO ADAPT	116
FIGURA 5-10 – PIPELINE SEQUENCIAL DE PROCESSAMENTO DA INFORMAÇÃO CONTEXTUAL.....	119
FIGURA 5-11 – MODELO DE CLASSES DA PIPELINE DE SUPORTE AO PROCESSAMENTO CONTEXTUAL.....	120
FIGURA 5-12 – PLATAFORMA XAMARIM.....	123
FIGURA 5-13 – DISTRIBUIÇÃO DOS DADOS DA AMOSTRA EM RESPEITO À ATIVIDADE (LOCKHART ET AL., 2011).....	127
FIGURA 5-14 – VARIAÇÃO DO SINAL NAS DIFERENTES ATIVIDADES CAMINHAR (À ESQUERDA) PARADO (À DIRETA).....	128
FIGURA 6-1 – ASPETO DO PORTAL SIDE UTILIZANDO UM SMARTPHONE	138
FIGURA 6-2 – REPRESENTAÇÃO GRÁFICA DO FLUXO DE INTERAÇÃO EM MODO NORMAL.....	142
FIGURA 6-3 – REPRESENTAÇÃO GRÁFICA DO FLUXO DE INTERAÇÃO EM MODO ENRIQUECIDO PELO CONTEXTO.....	143
FIGURA 6-4 – REPRESENTAÇÃO GRÁFICA DO FLUXO DE INTERAÇÃO DOS EVENTOS DE NOTIFICAÇÃO DE PROXIMIDADE DE UM DETERMINADO EVENTO.....	144
FIGURA 6-5 – PROCESSO DE DESENVOLVIMENTO E TESTE DE NOVAS FUNCIONALIDADES COM NOTIFICAÇÃO DE ERROS.....	146
FIGURA 6-6 – ASPETO DA APLICAÇÃO ALUNOSUTAD SEM CAPACIDADE CONTEXTUAL	149
FIGURA 6-7 – ASPETO DA APLICAÇÃO ALUNOSUTAD COM CAPACIDADE CONTEXTUAL NAS SITUAÇÕES EM QUE SE ENCONTRA A CAMINHO DA AULA, E DENTRO DA DISCIPLINA.....	150
FIGURA 6-8 – ASPETO DA APLICAÇÃO ALUNOSUTAD COM CAPACIDADE CONTEXTUAL NA SITUAÇÃO EM QUE O ALUNO SE ENCONTRA ATRASADO.....	151
FIGURA 6-9 – ASPETO DA APLICAÇÃO ALUNOSUTAD EM SITUAÇÃO DE TRANSIÇÃO DE DISCIPLINA	151
FIGURA 6-10 – ASPETO DAS NOTIFICAÇÕES ALUNOSUTAD EM SITUAÇÕES DE ALERTA IMPORTANTE	152
FIGURA 6-11 – PERFORMANCE OBTIDA PELA RANDOM FOREST COM VARIAÇÃO DO NÚMERO DE PROFUNDIDADE MÁXIMA E NÚMERO DE ÁRVORES = 10.....	154
FIGURA 6-12 – PERFORMANCE OBTIDA PELA RANDOM FOREST COM VARIAÇÃO DO NÚMERO DE ÁRVORES, COM PROFUNDIDADE MÁXIMA = 9	155
FIGURA 6-13 – GUIÃO DE AVALIAÇÃO DE TESTES DE ACEITAÇÃO DA APLICAÇÃO ALUNOSUTAD	157

LISTA DE TABELAS

TABELA 1-1 - FASES DA IMPLEMENTAÇÃO DA METODOLOGIA DE INVESTIGAÇÃO.....	10
TABELA 2-1 – TIPOS DE CATEGORIZAÇÃO CONTEXTUAL	24
TABELA 2-2 – COMPARAÇÃO DE CLASSIFICAÇÕES DE SENSIBILIDADE AO CONTEXTO	30
TABELA 2-3 - COMPARAÇÃO DE MÉTODOS DE AQUISIÇÃO BASEADOS NA ORIGEM	33
TABELA 2-4 – COMPARAÇÃO DAS DIFERENTES TÉCNICAS DE MODELAÇÃO CONTEXTUAL	38
TABELA 2-5 – RESUMO DE VÁRIOS PARAMETROS DE PROCESSAMENTO EM DOMÍNIO TEMPORAL.....	42
TABELA 2-6 – COMPARAÇÃO DAS DIFERENTES TÉCNICAS DE INFERÊNCIA CONTEXTUAL	45
TABELA 3-1 – RESUMO DO MIDDLEWARE EXISTENTE SENSÍVEL AO CONTEXTO	81
TABELA 4-1 – PRINCIPAIS DESAFIOS NO DESENVOLVIMENTO DE MIDDLEWARE SENSÍVEL AO CONTEXTO (ADAPTADO DE FERREIRA, 2013).	88
TABELA 5-1 – EXEMPLO DE EVENTOS PARA CADA TIPO DE FONTE CONTEXTUAL	115
TABELA 5-2 – TIPO DE EVENTOS CONSUMIDOS E PUBLICADOS PELA FONTE CONTEXTUAL LOCALIZAÇÃO INTERNA	125
TABELA 5-3 – ESPECIFICAÇÃO DA INFORMAÇÃO DE LOCALIZAÇÃO INTERNA	126
TABELA 5-4 – TIPO DE EVENTOS CONSUMIDOS E PUBLICADOS PELA FONTE CONTEXTUAL LOCALIZAÇÃO INTERNA	133
TABELA 5-5 – ESPECIFICAÇÃO DA INFORMAÇÃO DE LOCALIZAÇÃO INTERNA	133
TABELA 6-1 – RESUMO DE SERVIÇOS PRESTADOS PELO SIDE PARA OS ALUNOS (ADAPTADO DE BARBOSA ET AL., 2011).....	138
TABELA 6-2 – LISTA DE REQUISITOS FUNCIONAIS NÃO CONTEXTUAIS	140
TABELA 6-3 – LISTA DE REQUISITOS FUNCIONAIS CONTEXTUAIS	141
TABELA 6-4 – RESULTADO DOS TESTES DE RECONHECIMENTO DE ATIVIDADE HUMANA COM UMA CNN	156
TABELA 6-5 – RESULTADO DOS TESTES DE MANUAIS DE ACEITAÇÃO COM BASE NO GUIÃO PROPOSTO.....	157

ACRÓNIMOS

Nesta tese são utilizadas abreviaturas de designações comuns, apresentadas aquando da sua primeira utilização, na sua denominação original.

AP – Access Point

AIDL – Android Interface Definition Language

BSSID - Basic Service Set Identifier

BTS CID – Base Transceiver Station Cell Identifier

BLE – Bluetooth Low Energy

CNN – Convolutional Neural Network

ECA – Event - Condition - Action

ERM – Entity Relationship Model

GATT – Generic Attribute Profile

GPS – Global Positioning System

HCI – Human Computer Interface

HAR – Human Activity Recognition

HTTP – HyperText Transfer Protocol

ISO – International Organization for Standardization

JSON - JavaScript Object Notation

MQTT – Message Queuing Telemetry Transport

ORM – Object Role Model

QoC – Quality of Context.

RSS – Received Signal Strength

SDK – Software Development Kit

SMS – Short Messaging Service

UML – Unified Modelling Language

UUID – Universally Unique Identifier

USB – Universal Serial Bus

USB OTG – USB On-The-Go

XML – eXtensible Markup Language

[ESTA PÁGINA FOI INTENCIONALMENTE DEIXADA EM BRANCO]

1. INTRODUÇÃO

Neste primeiro capítulo é realizado um enquadramento do trabalho desenvolvido ao longo do desenvolvimento desta tese, apresentadas as motivações e objetivos para a sua realização, e definidas as suas principais contribuições. Este capítulo culminará com a apresentação da estrutura da tese, complementada com uma descrição resumida dos capítulos.

1.1. Enquadramento

A proliferação de dispositivos móveis transformou a forma de comunicarmos e interagirmos, alterando os nossos hábitos e permitindo o acesso a novas experiências, num mundo digital global. Estes dispositivos são considerados por muitos como as plataformas computacionais do futuro, a que não são alheias as estatísticas e previsões de penetração de mercado destes equipamentos, apresentando uma previsão de vendas perto dos 2 mil milhões de *smartphones* em 2020 (CCS Insight, 2016; Gartner Inc, 2016).

A utilização de dispositivos móveis alterou também as formas tradicionais de interação, desde a substituição dos habituais dispositivos de entrada, como rato e teclado, por ecrãs tácteis, até à utilização de reconhecimento de voz e processamento de linguagem natural, muitos têm sido os avanços tecnológicos, nesta área. A ubiquidade e consequente disponibilização de informação, em qualquer lado e qualquer lugar, é uma característica inerente aos dispositivos móveis, tornado o seu contexto de utilização altamente volátil. Entenda-se por contexto, não o seu sentido restrito, incluindo localização, ambiente, identidade, data, época e temperatura (Schilit, Adams, & Want, 1994; Brown, Chen, & Bovey, 1997), mas o sentido lato introduzido por Abowd (1999).

Este conjunto de fatores condiciona a utilização dos dispositivos móveis em certas situações, por limitações temporárias do utilizador e/ou pelas próprias condições do contexto (Magnusson et al., 2011). Complementarmente, os utilizadores com necessidades especiais são também condicionados pela complexidade da interface, por sobrecarga de aplicações e pelas condições do contexto. Consequentemente, a comunidade científica tem feito esforços no sentido de adequar os atuais modelos de interação à ubiquidade dos ambientes, propondo diversas soluções que asseguram o acesso universal das aplicações móveis em ambiente ubíquo. Algumas destas soluções recorrem a interfaces sensíveis ao contexto, com capacidade de adaptação da mesma às características do dispositivo (Hofer et al., 2003). Este nível de adaptação básico, não soluciona os problemas apresentados sendo, no entanto, amplamente

utilizado, nomeadamente ao nível de aplicações Web que, por exemplo, se adaptam às dimensões do dispositivo, mediante informação do cliente.

Recentemente, outras soluções recorrem a uma caracterização mais ampla do contexto para adaptação de interfaces em ambiente móvel contemplando, características do dispositivo, localização, eventos e frequência (e.g. Jiming, Chi Kuen, & Ka Keung, 2003; Kumar, Gupta, & Saha, 2006; Mowafi & Zhang, 2007).

Contudo, a aquisição do contexto não explora todas as potencialidades ao nível dos sensores dos dispositivos restringindo, conseqüentemente, a abrangência do conceito e a sua caracterização. Por outro lado, o alargamento do espectro do contexto poderá não assegurar a adaptação mediante as necessidades dos utilizadores, sendo tal evidenciado nos estudos de Lavie e Meyer (2010) e Paymans, Lindenberg e Neerincx (2004), em que são avaliados os impactos e *trade-offs* da adaptação de interfaces com base no contexto. A aquisição deste é também restrita ao nível dos sensores utilizados. A localização, bússola, acelerómetro e histórico de utilização, são as fontes de aquisição mais comuns, excluindo-se outros sensores, também por vezes utilizados, como a câmara e o microfone, ou mesmo a agenda do utilizador. Com a contextualização, é possível adaptar a interface, e a interação com a aplicação, à situação em que o utilizador está a utilizá-la e à ação que este pretende realizar, melhorando, assim, a sua usabilidade e, principalmente, a experiência de utilização.

1.2. Motivação

Os dispositivos móveis evoluíram muito desde a sua conceção, não sendo apenas utilizados para comunicar, mas tendo-se tornado fundamentais no dia-a-dia de muitos utilizadores (Böhmer, Hecht, Schöning, Krüger, & Bauer, 2011). Os dispositivos móveis aumentam as capacidades dos utilizadores com o acesso a informação, internet, música, navegação e entretenimento. Além destas capacidades avançadas dos dispositivos, estas são ainda melhoradas com aplicações desenvolvidas por terceiros, disponíveis através de um mercado digital aplicacional. No entanto, a forma como se interage com essas aplicações varia

muito consoante o nosso contexto, não só como as usamos, mas, quais usamos. Por exemplo, a forma e as aplicações com que interagimos é diferente se estamos a andar na rua, estamos em casa no sofá, a caminho do trabalho, ou numa reunião.

Esta inadaptabilidade das interfaces, em determinados momentos, torna a sua utilização desconfortável e, em alguns casos pode mesmo criar umas situações suscetíveis a acidentes. Nesta proposta, pretende-se desenvolver um modelo genérico que promova a adaptação dinâmica e sensível ao contexto de interfaces móveis. Algumas das questões colocadas por este problema passam por analisar de que forma os contextos influenciam a utilização das aplicações, de que forma é possível inferir um contexto com base nos sensores e fontes de informação do utilizador; de que forma é possível adaptar uma interface dinamicamente e, de que forma essa adaptação responde as necessidades do contexto interpretado.

Uma das questões fundamentais é, no entanto, como é possível construir uma interface com adaptação dinâmica ao contexto, de forma transparente para o utilizador e proporcionando uma experiência de utilização mais eficiente. Estas necessidades de adaptação do conteúdo ao utilizador, são alvo de investigação académica há algum tempo (e.g. Katz, 1994; Kindberg & Fox, 2004), tendo sido já apresentadas propostas de adaptação manual e dinâmicas como as propostas de Wai Yip e Lau (2002) e Narayanan, Flinn e Satyanarayanan (2000). A adaptação dinâmica é uma temática desafiante devido à natureza de utilização dos dispositivos móveis e ao seu uso, nos mais diversos contextos.

1.3. Tese

A utilização de aplicações móveis varia de acordo com o contexto em que se usam, assim, por vezes, durante a sua utilização, deparamo-nos com alguns obstáculos, que não são resolvidos pela aplicação num determinado contexto, e onde se a aplicação conseguisse proporcionar a adaptação correta tornaria a ação do utilizador mais eficiente. A tese aqui apresentada pode ser enunciada como:

A complexidade do processo de desenvolvimento de aplicações sensíveis ao contexto pode ser minimizada pela utilização de camadas de abstração intermédia, ou sistemas de middleware contextual, podendo estes sistemas ir além da partilha de informação, obtida com base em técnicas de inferência contextual ao permitir antecipar as necessidades de um utilizador, tendo em conta o seu contexto.

Esta tese tem associado um conjunto de questões de investigação (**QI**), orientadoras deste trabalho de investigação, apresentadas em seguida:

- QI-1.** Quais são os principais problemas na interação com as aplicações móveis passíveis de serem resolvidos com o contexto do utilizador?
- QI-2.** O que é o contexto do utilizador, quais são os contextos de utilização mais problemáticos, e de que forma é possível inferi-los programaticamente?
- QI-3.** De que forma é possível utilizar a informação contextual inferida de forma a beneficiar as aplicações móveis?
- QI-4.** Como é possível partilhar a informação contextual para múltiplas aplicações antecipatórias sensíveis ao contexto tendo em conta as restrições e limitações próprias de um equipamento móvel?
- QI-5.** Como é possível reduzir a complexidade do processo de desenvolvimento aplicacional provocado pela necessidade de implementação de técnicas de inferência?

1.4. Objetivos e contribuição

A construção da resposta deste desafio passa pela realização de várias etapas, necessárias para o enquadramento tecnológico do problema, e da escolha das soluções mais adequadas para a definição do modelo. Em seguida, serão apresentados os principais objetivos

parcelares, resultantes da principal questão desta tese, bem como as questões de investigação derivadas:

- Aprofundar e contribuir para o estado da arte com o conhecimento sobre contexto, sensibilidade ao contexto, e ciclo de vida contextual;
- Estudo de protocolos de comunicação e abstração de sensores físicos, de técnicas de inferência de dados contextuais, e arquiteturas para a criação de serviços isolados e independentes;
- Identificação das principais propostas de *middleware*, acadêmicas e comerciais, que promovam a sensibilidade ao contexto;
- Identificar as limitações na interação com as aplicações móveis passíveis de serem resolvidos com a informação contextual do utilizador.
- Proposta de um modelo inovador que permita a construção de um *middleware*, que dê resposta à sensibilidade contextual permitindo, assim, uma adaptação dinâmica e antecipatória da aplicação;
- Análise das tecnologias que possam servir de suporte à implementação de um *middleware* de inferência contextual antecipatória, baseada no modelo proposto;
- Implementação de um protótipo com a arquitetura proposta para validação do modelo;
- Realização de testes funcionais, qualitativos e quantitativos, de forma a avaliar a qualidade do modelo de inferência contextual.
- Disponibilizar uma camada de *middleware* que permita a reutilização da informação contextual.

A concretização destes objetivos parcelares possibilitará o desenvolvimento de um modelo que responde ao objetivo principal, através de uma arquitetura para um *middleware* para sensibilização contextual antecipatória (ADAPT), que ambiciona criar a fundação necessária para o desenvolvimento de aplicações móveis adaptativas contextuais.

Pretende-se com este trabalho contribuir para que o desenvolvimento de aplicações móveis possa ser mais ágil na implementação de sensibilidade contextual. Assim, é possível às aplicações móveis tirar maior proveito da informação contextual obtida a partir de um meio partilhado, permitindo que se adaptem e antecipem as necessidades do utilizador.

1.5. Metodologia

A investigação científica é definida como uma atividade que contribui para a compreensão de um determinado fenómeno que é, tipicamente, um conjunto de comportamentos ou entidades que apresenta relevância e interesse por parte da comunidade científica (Worrall & Currie, 1980; Kuhn & Hacking, 2012). Para esta contribuição ser válida, deve ser aceite pela comunidade de pares, através de divulgações científicas devendo, por isso, apresentar um elevado grau de novidade para essa comunidade (Gregor & Hevner, 2013). Assim, também as atividades realizadas durante a investigação devem ser alvo de escrutínio científico, pois apresentam uma grande relevância na qualidade do produto final, sendo designadas por metodologias de investigação.

Sendo o objetivo primordial deste trabalho, a elaboração de uma proposta, que tenha como âmbito a criação de conhecimento na forma de técnicas, métodos, modelos e teoria, a metodologia que mais se adequa é *design science*, a qual visa definir a investigação sob o ponto de vista de criação de artefactos, que permita efetuar o mapeamento entre os atributos e os requisitos funcionais. A investigação que usa a ciência para a construção de artefactos, é a mesma que cria o conhecimento requerido para essa construção, através de técnicas de design, análise, reflexão e abstração (Hevner & Chatterjee, 2010).

De uma forma geral, a investigação segundo a metodologia *design science*, pode ser representada pela seguinte figura:

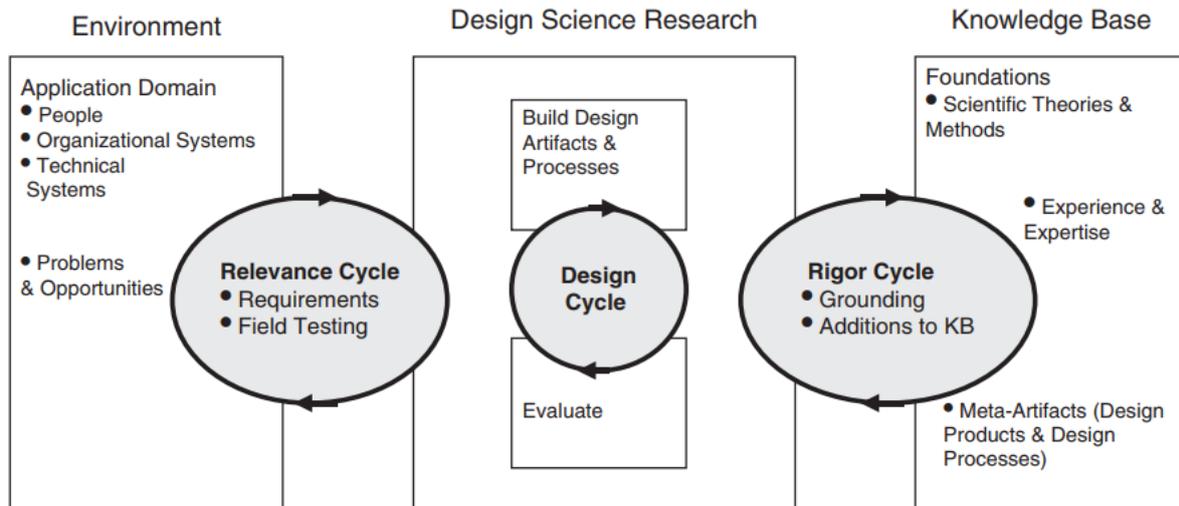


FIGURA 1-1 – MODELO GERAL DAS FASES DE IMPLEMENTAÇÃO DA METODOLOGIA DE INVESTIGAÇÃO BASEADA EM DESIGN SCIENCE (ADAPTADO DE HEVNER & CHATTERJEE, 2010).

O modelo apresentado na Figura 1-1, realça a contribuição do novo conhecimento como um facto chave na aplicação da metodologia sendo que, de uma forma geral, o processo segue as seguintes fases:

Consciência do problema – percepção da origem do problema, como as necessidades ou propostas existentes. Esta etapa possui como resultado uma proposta para a atividade de investigação.

Sugestão – é uma fase criativa onde se começa a planificar o artefacto, com base nos elementos existentes ou com vista aos elementos a serem criados.

Desenvolvimento – é o desenvolvimento do projeto propriamente dito, que deve seguir métodos e técnicas de estado da arte. A novidade e a contribuição para o conhecimento científico devem estar nos aspetos do artefacto e não nas técnicas de construção e desenvolvimento.

Avaliação – a avaliação dos artefactos, de forma qualitativa e quantitativa, com vista à comparação com os pressupostos do que era esperado na fase de proposta inicial. Esta fase tem como missão confirmar e contradizer as hipóteses.

Conclusão – fase final do ciclo de investigação, que ocorre tipicamente quando todos os períodos de investigação se encontram satisfatoriamente atingidos, ou seja, quando os resultados obtidos são suficientemente bons, apesar da possibilidade de poderem ter ocorrido diversos desvios e alterações à proposta inicial.

Contribuição de conhecimento (Divulgação) – validação do trabalho efetuado por pares, a qual necessita de detalhe suficiente para permitir a replicação do artefacto proposto e implementado.

Face ao exposto, a metodologia apresentada agrupa uma sequência de fases distintas, as quais se repetem, de forma iterativa, levando ao aperfeiçoamento do artefacto final com o conhecimento das avaliações anteriores. A Tabela 1-1, resume a forma de como o trabalho de investigação proposto corresponde à metodologia de investigação *design science*.

TABELA 1-1 - FASES DA IMPLEMENTAÇÃO DA METODOLOGIA DE INVESTIGAÇÃO

ETAPA	DESCRIÇÃO
CONSCIÊNCIA DO PROBLEMA	DEMONSTRAR E VALIDAR UM MODELO DE QUE PERMITA A INFERÊNCIA CONTEXTUAL DO UTILIZADOR E APLICAÇÃO DESSE CONHECIMENTO DE FORMA A AUMENTAR A EFICÁCIA E EFICIÊNCIA DO UTILIZADOR, NA UTILIZAÇÃO DE APLICAÇÕES MÓVEIS.
SUGESTÃO	PROPOR ATRAVÉS DA COMBINAÇÃO DE DIVERSAS FONTES SENSORIAIS E TÉCNICAS DE CLASSIFICAÇÃO E INFERÊNCIA CONTEXTUAL PARA DETERMINAR O CONTEXTO PARA PERMITIR ADAPTAR CORRETAMENTE AS INTERFACES NECESSIDADES DO UTILIZADOR.
DESENVOLVIMENTO	UTILIZAR UM CENÁRIO REAL PARA VALIDAR OS MODELOS PROPOSTO, ATRAVÉS DA CONSTRUÇÃO DE UM PROTÓTIPO QUE PERMITA IDENTIFICAR UM CONJUNTO DE SITUAÇÕES E ADAPTAR-SE CORRETAMENTE ÀS MESMAS.
AValiação	AVALIAR O CUMPRIMENTO DAS ESPECIFICAÇÕES PROPOSTAS NA SUGESTÃO POR PARTE DO PROTÓTIPO. ESTA FASE É CONSTITUÍDA POR VÁRIAS CATEGORIAS DE TESTES, SENDO A FASE FINAL A AVALIAÇÃO POR UTILIZADORES REAIS EM AMBIENTE REAL.
CONCLUSÃO	ESTA FASE SERÁ ATINGIDA QUANDO A SOLUÇÃO PROPOSTA SE CONSIDERAR SUFICIENTEMENTE SÓLIDA E CORRESPONDER NA TOTALIDADE AOS OBJETIVOS PROPOSTOS.
CONTRIBUIÇÃO DE CONHECIMENTO	SERÁ REALIZADA SOB A FORMA DE PUBLICAÇÕES CIENTÍFICAS PERIÓDICAS E SOB A FORMA DO PRESENTE DOCUMENTO.

1.6. Organização da tese

Além do atual capítulo que apresenta o enquadramento, a motivação, a tese, os objetivos e contribuições e metodologia, a presente tese apresenta ainda um conjunto de seis capítulos organizados da seguinte forma:

Capítulo 2 – Conceitos de contexto

Este capítulo apresenta e contextualiza a temática da adaptação dinâmica sensível ao contexto em dispositivos móveis. Apresentam-se as bases científicas que servem de pilares para a investigação a realizar identificando, assim, a definição de contexto e de sensibilidade ao contexto que nos serviram de guia para a presente investigação. É também apresentada uma revisão sobre o ciclo de vida contextual e as técnicas utilizadas, em cada fase, para o processar.

Capítulo 3 – Estado da arte de middleware contextual

Este capítulo descreve as principais propostas, ao nível da sensibilidade ao contexto, existentes na literatura para as diferentes fases de vida do contexto de sistemas de middleware, realizando uma comparação e tecendo algumas conclusões.

Capítulo 4 – Modelo contextual antecipatório

Neste capítulo é apresentado, o ADAPT, modelo de adaptação dinâmica e sensível ao contexto de interfaces móveis em ambientes ubíquos. Este capítulo é introduzido pela apresentação das características genéricas de um sistema de *middleware* sendo, em seguida, apresentadas os principais desafios e diretrizes orientadoras utilizadas para o desenvolvimento desta tese. Por fim, neste capítulo, é descrito de forma introdutória o modelo de adaptação contextual para a construção de aplicações móveis antecipatórias.

Capítulo 5 – Arquitetura

Este capítulo contém a apresentação, descrição e detalhes da implementação da arquitetura de suporte ao modelo contextual antecipatório ADAPT, desenvolvido para implementar o modelo proposto e permitir a sua validação. Este capítulo começa com a apresentação das tecnologias de suporte à arquitetura, sendo de seguida iniciada a apresentação da arquitetura, começando por ser dada uma perspetiva genérica do sistema, e sendo, progressivamente, pormenorizado cada um dos seus componentes até ao nível da sua implementação.

Capítulo 6 – Caso de estudo e avaliação

Neste capítulo é apresentada a avaliação e validação do ADAPT e da arquitetura de suporte desenvolvida, suportada pela apresentação do caso de estudo que serviu de base à realização de experiências com utilizadores reais. A avaliação da arquitetura é apresentada sob uma vista qualitativa ao nível da interação e quantitativa ao nível do desempenho e precisão do sistema.

Capítulo 7 – Conclusões e trabalho futuro

No capítulo final, é apresentada uma síntese dos pontos mais relevantes do trabalho desenvolvido, sendo realizada uma análise crítica das características do modelo proposto e a avaliação do estado atual do ADAPT. São tecidas algumas conclusões sobre o trabalho, com particular destaque para a validação dos objetivos propostos inicialmente. Serão também exploradas potenciais linhas de investigação futuras, de forma a permitir a continuidade do trabalho realizado.

2. CONCEITOS DE CONTEXTO

Neste capítulo, será apresentada a definição de contexto no âmbito da aplicação a dispositivos móveis e os seus principais desafios, incluindo a definição de sensibilidade ao contexto. Será analisado o ciclo de vida do contexto em dispositivos móveis e as principais propostas em cada uma das fases. Este capítulo termina com uma revisão sobre algumas técnicas de inferência contextual.

2.1. Enquadramento

Os *smartphones* evoluíram muito desde a sua invenção começando, inicialmente, por possuírem apenas a capacidade de realizar chamadas e evoluindo para os dias de hoje, em que são autênticos computadores e assistentes pessoais, que conseguem interpretar comandos em linguagem natural. Este aumento de capacidades e funcionalidades dos equipamentos móveis torna-os cada vez mais indispensáveis no nosso dia-a-dia.

Uma das características que se destaca nesta evolução é o aumento de tamanho do ecrã dos dispositivos. Esta evolução deu-se também noutras características como o tipo e qualidade de ecrã e também na forma como interagimos com este. Estas características estão permanentemente a evoluir, sendo possível, hoje em dia, encontrar dispositivos móveis de todos os tamanhos e formatos.

Com o aumento do tamanho do ecrã, a criação de novas formas de interação com os dispositivos e a diversidade de aplicações que o dispositivo possui, o contexto de utilização do dispositivo adquire especial relevância. Mesmo uma aplicação que é bastante eficaz em determinadas situações, pode torna-se frustrante de utilizar, quando a usamos noutra situação do nosso dia-a-dia. Por vezes, quando necessitamos de ler um email ou SMS, com urgência, enquanto nos dirigimos para um determinado local, somos confrontados com a dificuldade de utilização do dispositivo, que nos impede de encontrar a aplicação ou de ler corretamente a mensagem. Outras vezes, as aplicações são desenhadas para serem utilizadas em situações em que temos a necessidade de nos concentrar, o que frustra ou impossibilita o seu uso, por exemplo, quando estamos em movimento.

Os equipamentos móveis atuais possuem a capacidade de inferir se os seus utilizadores se encontram em movimento ou em corrida (Miluzzo et al., 2008), a dormir ou a deslocarem-se para o trabalho (Nicholas et al., 2011), ou mesmo, qual o estado emocional em que o utilizador se encontra (Musolesi, Rachuri, & Mascolo, 2011). Hoje em dia, uma grande parte da população possui um *smartphone* que, na maior parte dos casos, está intimamente integrado

no dia-a-dia das pessoas. Para além da sua presença constante, também auxiliam em tarefas como: organização de reuniões e encontros, navegação, pesquisa de informação e locais, *networking* e comércio eletrónico adaptando-se à nossa forma de utilização e reestruturam e reorganizam-se, de forma a apresentarem-nos uma interface eficiente para a situação em que nos encontramos.

De forma a propor uma solução para o problema, é necessário compreender a complexidade e dimensão do problema. Assim, é fundamental compreender corretamente o que é o contexto, a sensibilidade ao contexto, como é possível inferir e representá-lo e, por fim, analisar de que forma tudo isto pode ser utilizado para criar uma melhor experiência de utilização. Para além desta melhoria, é possível ainda dar mais um passo, na direção da antecipação das necessidades do utilizador e tornar, assim, o resultado final mais eficaz e eficiente, tornando-o uma mais-valia para o utilizador.

Os sistemas antecipatórios reforçam as capacidades de sensorização móvel e as técnicas de inferência com a capacidade de prever eventos futuros. Os sistemas antecipatórios são uma área multidisciplinar, que interliga a sensorização móvel, interação pessoa-computador (HCI), *machine learning* e inferência de contexto. Os sistemas antecipatórios, tal como propostos por Rosen:

“An anticipatory system is a natural system that contains an internal predictive model of itself and of its environment, which allows it to change state at an instant in accord with the model’s predictions pertaining to a later instant.” (Rosen, 2012)¹

Rosen sugere que um sistema antecipatório necessita de ser capaz de obter uma imagem realista do seu estado e do seu meio envolvente, isto é, o contexto do utilizador e do equipamento. A construção bem-sucedida de um sistema antecipatório passa por dois requisitos: a presença de um conjunto de hardware que consiga inferir e modelar o futuro da ação; e a sua integração no quotidiano do utilizador.

¹ Tradução Livre do Autor - Um sistema que contém um modelo preditivo de ele próprio e do seu ambiente que permite alterar o seu estado num instante de acordo com as predições do modelo relativas a um instante posterior (Rosen, 2012)

Os *smartphones* apresentam-se como equipamentos fundamentais ao nível da construção de sistemas antecipatórios móveis, fazendo a ponte entre os dispositivos eletrónicos e o utilizador. Estes equipamentos cumprem o primeiro requisito, predição antecipatória, possuindo sensores que conseguem inferir e monitorizar o contexto e, também, hardware que é capaz de processar diversos algoritmos de inferência, que permitem construir diversos modelos do futuro da ação do utilizador. Em segundo lugar, estes equipamentos encontram-se integrados de forma bastante profunda, na vida e dia-a-dia dos utilizadores, permitindo assim que os modelos inferidos pelo equipamento sejam pessoais, oportunos e relevantes para o utilizador. Estes equipamentos podem também inferir os padrões comportamentais do utilizador e apoiá-lo, com a apresentação de uma interface que seja adaptada às necessidades do utilizador. A revisão descrita neste capítulo teve em conta a aplicação da seguinte metodologia:

1. Construção de uma base de dados de propostas científicas utilizando os seguintes parâmetros “*Context-Aware**”, “*Ubiquitous computing*”, “*middleware OR framework*”, “*mobile*”, utilizando as bases de dados Web of Knowledge (<http://webofknowledge.com>), Science Direct (<http://www.sciencedirect.com/>), IEEE Explore (<http://ieeexplore.ieee.org/>), e Google Scholar (<https://scholar.google.pt/>). Os resultados foram importados para o programa *Publish or Perish* (Harzing, 2007), uma aplicação gratuita que permite obter métricas sobre os artigos de forma simples, nomeadamente, ao nível de citações, total, ajustado ao ano, e deteção automática de duplicados, entre outras. Após a limpeza de duplicados compilou-se uma lista de 1326 artigos, dos quais foram removidos 372 por não possuímos acesso ao documento.
2. Com base na lista de 954 artigos, procedemos à exclusão de propostas por análise dos títulos e resumos dos artigos, e em caso de dúvida, por análise do artigo, tendo sido removidos os artigos de tipo: *Surveys não relativos à análise da temática contextual (67)*, *apenas sobre a aplicação ou middleware sensível ao contexto (416)*, *não relacionadas com contexto (173)*. Após esta análise, obteve-se uma lista de 298 artigos.

Nesta secção é apresentada, em detalhe, a complexidade existente na definição e especificação de contexto consoante as suas aplicações. Em seguida é apresentada a visão da sensibilidade ao contexto, que é o conceito base da utilização de contexto para o melhoramento das aplicações. Posteriormente, é apresentada uma análise sobre o ciclo de vida do contexto quando aplicado na área tecnológica, no domínio das aplicações móveis, incluindo a apresentação das tecnologias de suporte relativas à inferência contextual da informação contextual complexa. Por fim, este capítulo será concluído com a apresentação de aplicações de estado da arte relativamente aos desafios da inferência contextual para duas situações a localização interna em edifícios e o reconhecimento de atividade humana.

2.2. Contexto

Nesta secção será apresentada uma revisão das várias propostas de definição de contexto nas diferentes áreas científicas. Esta revisão começa pela identificação da importância do conceito de contexto, passando em seguida para a apresentação de diferentes propostas de definição de contexto nas diferentes áreas científicas. A definição de contexto converge na área das ciências da computação, onde se apresenta a evolução na construção da definição de contexto até concluir com a definição que nos irá guiar nesta tese. Posteriormente, é apresentada uma revisão sobre as diferentes propostas de categorização contextual, apresentando as suas diferenças e a convergência numa que complementa as existentes.

2.2.1. Introdução

O termo contexto é algo que, à partida, pode parecer simples, mas, à medida que refletimos sobre o tema, a sua definição torna-se cada vez mais complexa. O contexto possuiu uma elevada importância, pois depende dele a interpretação e o significado que se atribui à informação. Um bom exemplo disso é a área de Inteligência e Segurança, onde Heuer, num dos seus mais conhecidos trabalhos na área de análise de informação de inteligência, descreve que *a importância da informação é sempre uma reunião entre a natureza da informação e o contexto em que a mesma é interpretada* (Heuer & Center for the Study of Intelligence - CIA, 1999). Heuer, descreve que, o contexto tem importância na obtenção da informação e na forma como esta é utilizada ou interpretada, podendo mesmo alterar a sua importância ou significado.

Dependendo da área científica, a definição de contexto pode ser diferente. A diversidade de definições é de tal forma complexa, que os investigadores, Bazire e Brézillon (2005), examinaram 150 definições de contexto, de entre as mais diversas áreas de investigação, concluindo a sua análise afirmando que a criação de uma definição unificada de contexto é um esforço árduo e provavelmente, impossível. Nesse trabalho, os autores concluem que a definição de contexto depende principalmente do campo em que a mesma está a ser aplicada,,

mas que, de uma forma geral, nas definições analisadas, o contexto é tido como um conjunto de restrições que influenciam o comportamento relativo a uma determinada tarefa (Bazire & Brézillon, 2005).

A discussão desta temática apresenta uma relevância tal, que diversos especialistas se reúnem para participar numa conferência interdisciplinar, a CONTEXT, que terá a sua décima edição em junho de 2017, para discutir este tópico e as suas aplicações nos diversos ramos da ciência, engenharia, direito, psicologia, filosofia e outros. A definição de contexto tem sido alvo de diversa discussão, desde a área da psicologia (Dijk, 2008), a linguística (Requejo, 2007), e até às diversas áreas da informática, onde possuiu diversas diferenças, tais como as destacadas por O'Connor, Cunningham e Cahill (2007), Di Sciullo (1999), Ekbia e Maguitman (2001) e Julien, Payton e Roman (2004). A definição de contexto é dependente do contexto em que o mesmo é utilizado e, como tal, a sua definição específica varia consoante a área em que este é utilizado.

Na vertente das ciências cognitivas, Kokinov (1999) apresentou uma proposta, a teoria dinâmica de contexto, onde o contexto se define como sendo uma reunião de todas as entidades que influenciam o comportamento humano ou de um sistema. A teoria dinâmica de contexto de Kokinov assenta em quatro princípios: (1) o contexto é um estado mental, (2) o contexto não possui limites claros, (3) o contexto consiste em todos os elementos que são relevantes, (4) o contexto é dinâmico (Kokinov, 1999).

O contexto é também alvo de análise noutras áreas, especialmente na Inteligência artificial, com uma abordagem relativamente formal, proposta por Lenat (1998), que sugere que o contexto pode ser descrito como “*uma região num espaço n-dimensional*”.

Ao nível do raciocínio lógico, McCarthy e Buvac (1997), um dos trabalhos mais conhecidos nesta temática, exploram uma formato de raciocínio lógico, onde as proposições e os termos são afirmados num contexto específico com base nas relações $ist(c,p)$ (em que a proposição p é verdadeira no contexto c), $value(e,c)$ (definindo o valor e em contexto c). Neste trabalho, os autores apresentam algumas reflexões sobre o contexto: (1) o contexto é sempre

relativo a outro contexto, (2) o contexto é infinito, (3) não é possível descrever o contexto na sua totalidade, (4) numa situação de diversos contextos distintos, existe sempre um contexto global superior, onde os mesmos podem ser mapeados na totalidade.

Existe uma clara divergência de opiniões entre as abordagens da engenharia e as das ciências humanas e sociais. Com o objetivo de comparar estas áreas, Dourish (2004) tenta apresentar no seu trabalho uma comparação entre as diversas áreas, comparando a vertente positivista, que é típica em engenharia, e a vertente fenomenológica, que é mais utilizada nas áreas das ciências humanas e sociais.

Na vertente positivista, o contexto é visto como uma fonte de informação, uma entidade que é estável, delineável e separável da atividade. O autor indica que, segundo esta vertente, o contexto é visto como sendo um problema de representação. Em oposição, na vertente fenomenológica, o contexto é visto como uma propriedade da relação entre os objetos ou atividades associadas com ações e participantes específicos. Nesta visão o contexto é considerado como produto da atividade, logo não sendo separável da mesma. Segundo esta vertente, o contexto é visto como um problema interacional (Dourish, 2004).

Dourish propõe ainda a utilização da visão fenomenológica, ao invés da visão clássica de engenharia, a positivista nas futuras aplicações de contexto, mesmo nas aplicações de engenharia. Esta proposta deve-se às vantagens que a abordagem fenomenológica permite pois, como se baseia num problema interacional, permitiria suportar, não só as atividades do utilizador, mas também a sua evolução ao longo do tempo, algo que não é possível quando o problema é considerado apenas como problema de representação e codificação (Dourish, 2004).

2.2.2. Revisão das definições de contexto

Contexto é um conceito complexo. Face à sua complexidade, tomemos como ponto de partida a definição genérica do termo, constante no dicionário Merriam-Webster's Collegiate:

“Contexto é a situação em que acontece alguma coisa, ou o grupo de condições que existiam onde e quando alguma coisa acontece”.

Genericamente, contexto é um conjunto de informação que ajuda a compreender a situação e que pode ajudar a reagir à situação se a reação for, de facto, necessária (Devlin, 2005).

Como já exposto, o conceito de contexto tem sido alvo de várias exposições por diversos autores o que permite, assim, construir uma base de conhecimento, de senso comum e conhecimento comum, sobre a temática. Lenat, Shepherd, Guha, Pittman e Pratt (1990), descrevem de forma detalhada, o conceito com base no conhecimento humano e senso comum, onde o contexto é descrito através de 12 dimensões que consideram as mais importantes para caracterizar os tipos gerais de conhecimento, que incluem, entre outras, o tempo, local, cultura, granularidade e a justificação.

Na área da ciência da computação, a abordagem ao tema de sensibilidade ao contexto é apresentado pela primeira vez por Schilit e Theimer (1994), onde este é representado por elementos caracterizadores como: a localização, as entidades das pessoas próximas e os objetos e as alterações dos mesmos.

Uma definição mais detalhada é formulada, posteriormente, por Schilit et al. (1994), no qual o contexto inclui as categorias (1) contexto computacional, (2) contexto utilizador, (3) contexto físico. Outros investigadores enriqueceram a definição inicial proposta por Schilit et al. (1994), mas de forma geral apenas acrescentavam as variáveis que necessitavam para a sua investigação. Isto verifica-se, por exemplo, nos trabalhos de Brown, que acrescentou o conceito temporal (Brown et al., 1997) e de Dey que acrescentou o estado emocional e a atenção do utilizador (Dey, Wood, & Abowd, 1998; Dey, Futakawa, Salber, & Abowd, 1999).

A definição de contexto foi sendo enriquecida à medida que os investigadores necessitavam de mais uma variável para o seu trabalho e as definições anteriores existentes não se enquadravam ao que se pretendia. Devido a esta divergência de definições, surgiu a necessidade de criar uma definição que fosse flexível aos crescentes requisitos, mas que, no

entanto, fosse descritiva o suficiente para o problema. Essa necessidade levou à definição proposta por Anind Dey:

“Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.” (Dey, 2001)²

Esta definição defende que a ação do utilizador é caracterizada como sendo também ela própria um objeto do contexto, visão partilhada por outros autores (e.g. Crowley, Coutaz, Rey, & Reignier, 2002; Cassens & Kofod, 2006). Em trabalhos posteriores, como Chen (2003), este estende a definição de contexto às atividades e ações que estão a ocorrer numa determinada localização.

Esta definição tem sido uma das mais usadas atualmente, mesmo noutros campos, como na vertente da operacionalização. Zimmermann, Lorenz e Oppermann (2007) partem inicialmente da proposta apresentada por Dey, desconstruindo-a nos seus cinco elementos base: (1) *Individualidade* (propriedades e atributos da entidade), (2) *Atividade* (todas as tarefas que a entidade está envolvida), (3) *Localização* (área ou local onde a entidade está ou pretende ir), (4) *Tempo* (relativa à ação pretendida) e (5) *Relações* (informações que a entidade por estabelecer com outras entidades).

Apesar da generalidade da definição, proposta por Dey (2001), ela apresenta algumas lacunas, especialmente ao nível das fronteiras entre alguns conceitos, especialmente entre *contexto*, *modelo contextual*, e *informação contextual*.

Henricksen (2003), caracteriza a informação contextual como um mecanismo que possibilita a realização de ações pelo utilizador de forma automatizada e flexível. Para esta autora, o contexto é considerado em relação às ações entre o utilizador e as aplicações. Assim, propõe a sua interpretação para os seguintes conceitos, Contexto, Modelo contextual, e

²Tradução livre do autor - “Contexto é qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, um local ou um objeto que possa ser considerado relevante para a interação entre o utilizador e a aplicação incluindo o utilizador e aplicação.” (Dey, 2001)

Informação contextual: o contexto é considerado ao nível da ação, e desse modo propõe que o contexto de uma ação é o conjunto de circunstâncias que a envolvem e que são de potencial relevância para a concluir.

Ao nível do modelo contextual, ele é proposto como sendo um subconjunto concreto do contexto, que é realisticamente capaz de ser obtido a partir dos sensores, aplicações, e utilizadores, e que pode ser explorado para a execução de uma ação. O modelo contextual é geralmente aplicado por uma aplicação sensível ao contexto de forma explícita, conforme programado na aplicação mas a mesma pode evoluir com o tempo (Henricksen, 2003).

Por fim, define atributo contextual como sendo um conjunto de dados obtidos dos sensores, que pertence ao modelo contextual. Desta forma, o atributo contextual providencia uma amostra, de um determinado ponto no tempo, do subconjunto do contexto (Henricksen, 2003). A caracterização contextual é de elevada importância para ajudar a compreender, utilizar e interpretar os diversos tipos de contextos de uma maneira eficiente.

As categorizações estão intimamente relacionadas com a forma como o contexto é interpretado e as limitações que impõe sobre a informação. A Tabela 2-1, apresenta uma revisão das diversas visões de categorização contextual apresentadas pelos vários autores, a partir da metodologia descrita na secção de enquadramento deste capítulo, analisando os resultados obtidos pela aplicação das duas primeiras etapas de seleção.

TABELA 2-1 – TIPOS DE CATEGORIZAÇÃO CONTEXTUAL

TIPO DE CATEGORIZAÇÃO	DESCRIÇÃO
WHO, WHEN, WHERE, WHAT, WHY, HOW (Kim, Baik, & Son, 2012)	ANÁLISE DA SITUAÇÃO E DO AMBIENTE DE DIFERENTES PONTOS DE VISTA.
FÍSICO / HUMANO (Schmidt, 2002)	DIFERENCIAÇÃO COM BASE NA ORIGEM: HUMANOS (UTILIZADOR, AMBIENTE SOCIAL, TAREFAS), FÍSICAS (LOCALIZAÇÃO, INFRAESTRUTURA, CONDIÇÕES)

<p>ESTÁTICO/DINÂMICO (Nadales, Röckl, & Frank, 2010)</p>	<p>DIFERENCIAÇÃO PELA OBSERVAÇÃO AO LONGO DO TEMPO: NÃO VARIÁVEL (ESTÁTICO), ADAPTÁVEL ÀS ALTERAÇÕES (DINÂMICO)</p>
<p>PRIMÁRIO / SECUNDÁRIO (Abowd et al., 1999; Perera, Zaslavsky, Christen, & Georgakopoulos, 2014) DIRETO / INDIRETO (Prado & Ortiz, 2011)</p>	<p>DIFERENCIAÇÃO ATRAVÉS DA COMPLEXIDADE DE OBTENÇÃO; CONTEXTO INDIRETO/SECUNDÁRIO É MAIS COMPLEXO DE OBTER POIS NECESSITA DE PROCESSAMENTO, INFERÊNCIA, ETC. O CONTEXTO DIRETO É OBTIDO DIRETAMENTE DOS SENSORES</p>
<p>SENSORIAL, COMBINADO, INFERIDO, APRENDIDO (Lee, Kim, Lee, & Lee, 2007)</p>	<p>DIFERENCIAÇÃO ATRAVÉS DA COMPLEXIDADE DE OBTENÇÃO DE FORMA MAIS CATEGORIZADA. <i>SENSORIAL</i> – DIRETO DOS SENSORES, <i>COMBINADA</i> – INFORMAÇÃO OBTIDA DE VÁRIOS SENSORES, <i>INFERIDA</i> – OBTIDA POR REGRAS, <i>APRENDIDA</i> – COMPARADA COM SITUAÇÕES SEMELHANTES.</p>
<p>SENSORIAL, ESTÁTICA, PERFIL, DERIVADA (Henricksen, 2003)</p>	<p>DIFERENCIAÇÃO ATRAVÉS DA COMPLEXIDADE DE OBTENÇÃO DE FORMA MAIS CATEGORIZADA. <i>SENSORIAL</i> – DIRETA DOS SENSORES; <i>ESTÁTICA</i> – INFORMAÇÃO NÃO ALTERÁVEL; <i>PERFIL</i> – INFORMAÇÃO VARIÁVEL DE BAIXA FREQUÊNCIA; <i>DERIVADA</i> – OBTIDA PELO PROCESSAMENTO DOS SENSORES</p>
<p>INTERNO / EXTERNO (Gwizdka, 2000)</p>	<p>CATEGORIZAÇÃO COM BASE NO PONTO DE VISTA DO UTILIZADOR. INTERNOS – INICIADO PELO UTILIZADOR; EXTERNO – QUE INFLUENCIAM O UTILIZADOR</p>

2.2.3. Definição de contexto

Sendo o aumento da eficiência e eficácia do utilizador, uma das nossas principais questões de investigação, a nossa definição de contexto centra-se em volta da ação do utilizador, pois, assim, a correta adaptação da aplicação depende do resultado que o utilizador pretende. Por este motivo, a definição de contexto utilizada no nosso trabalho segue a definição de Dey (2001), mas enriquecida pela informação da ação que o utilizador pretende realizar, proposto por Henricksen (2003), de forma a permitir tornar o sistema antecipatório. Assim, nesta tese, consideramos contexto como sendo:

“Qualquer tipo de informação e circunstâncias relevantes, que possam ser utilizadas para representar a ação que uma determinada entidade pretende realizar e as restrições que a mesma sofre em um determinado momento, incluindo as ações anteriores relevantes.”

Nesta definição, considera-se que: a *ação* é o que se pretende obter quando a execução é bem-sucedida; a *entidade* pode ser um utilizador, uma aplicação, ou um serviço autónomo; as *restrições* são todas as características que impossibilitam ou prejudicam a realização da ação de forma correta, e o *momento* é a duração em que essas restrições se aplicam à ação que se pretende realizar, podendo ser um instante, ou prolongada no tempo.

Assim o contexto pode ser visto, de uma forma reduzida, como sendo a interceção da entidade com a ação pretendida e as características que podem prejudicar e favorecer essa ação, num determinado momento. Deste modo, o contexto é enriquecido também com a informação das ações do passado, do presente e com a predição do resultado no futuro. Definem-se, assim, os requisitos para o desenvolvimento de um sistema antecipatório.

No desenvolvimento do nosso modelo será ainda seguida uma caracterização contextual, inspirada pelas propostas de Henricksen (2003) e Lee et al. (2007). Propomos, nesse sentido a seguinte categorização:

Sensorial – Toda a informação que possuiu uma taxa de alteração alta, sendo normalmente obtida de sensores físicos, reais ou virtuais, incluindo a informação obtida pelo processamento simples da informação dos sensores e ou do conjunto de múltiplos sensores.

Estática – Toda a informação que nunca varia, ao longo do tempo de vida do dispositivo, independentemente do contexto, mas que pode variar de dispositivo para dispositivo. Esta pode incluir informação relativa às características e limitações dos sensores, entre outros.

Perfil – Toda a informação que possui uma taxa de alteração muito baixa, que pode incluir a informação da entidade nomeadamente, preferências da entidade, entre outras.

Inferida – Toda a informação que foi obtida pela aplicação de técnicas de inteligência artificial, obtida por regras, ou por outros algoritmos mais complexos, recolhida a partir de todas as outras fontes de informação. Pode incluir toda a informação relevante que foi

previamente obtida na realização da ação anterior, que inclui a taxa de eficácia da ação e as restantes fontes de informação no momento da ação. (Sensorial, Estática, Perfil, Aprendida)

2.2.4. Considerações finais

Nesta secção, foi analisado o contexto, a importância do mesmo e a dificuldade em o definir de uma forma geral. Analisaram-se as várias propostas de diversos autores, apresentando-se, deste modo, uma evolução no enriquecimento da definição, até obtermos a proposta de contexto que nos irá guiar nesta tese.

Foram também apresentados, nesta secção, alguns dos diferentes tipos de caracterização contextual, elencando-se, de forma resumida, as principais diferenças entre si, e concluindo com a nossa interpretação de contexto, que servirá de suporte à implementação do nosso modelo.

2.3. Sensibilidade ao contexto

De forma a ser possível utilizar o contexto é necessário perceber o que é, efetivamente, ser sensível ao contexto. Por este motivo, nesta secção será apresentada uma revisão sobre a sensibilidade ao contexto, que visa fundamentar e enriquecer a compreensão da temática.

2.3.1. Introdução

A sensibilidade ao contexto é reconhecida como sendo parte da computação ubíqua. A computação ubíqua é um conceito introduzido por Weiser (1999), apresentando o conceito da tecnologia “*anytime, everywhere, any device, any location and in any format*”. A sensibilidade ao contexto é muito semelhante à computação senciente, tendo como principal diferença, a circunscrição à utilização de sensores físicos para a inferência de contexto, no caso da computação senciente (Addlesee et al., 2001).

O foco na sensibilidade ao contexto evoluiu, nas últimas décadas, das aplicações de desktop, aplicações web, aplicações móveis, computação ubíqua/perversiva, até à Internet of Things (Perera et al., 2014).

2.3.2. Sensibilidade ao contexto

A sensibilidade ao contexto é uma visão da computação centrada no utilizador, na qual, o sistema computacional se adapta dinamicamente e de forma não intrusiva, às necessidades do utilizador no ambiente em que se encontra. Por este motivo, uma aplicação sensível ao contexto pode apresentar diferentes semânticas, diferentes formatos, diferentes tipos de ações e informação dependendo do contexto em que está a ser utilizada.

O conceito de sensibilidade ao contexto aplicacional foi apresentado pela primeira vez por Schilit e Theimer (1994), onde referem que, as aplicações são sensíveis ao contexto quando

elas próprias se conseguem adaptar ao contexto. De forma semelhante à definição de contexto, também este conceito divergiu consoante as necessidades dos autores e, por esse motivo, surgiu a necessidade de forma construir uma definição unificada.

Pascoe (1998) formulou que a sensibilidade ao contexto é a capacidade dos dispositivos de detetar, interpretar e reagir, às alterações do ambiente local do utilizador e dos próprios dispositivos. Outros autores apresentam definições semelhantes, que tratam a sensibilidade ao contexto como sendo a capacidade de se adaptar ou alterar dinamicamente a sua forma de atuar, de acordo com o estado da aplicação ou do utilizador (e.g. Schilit et al., 1994; Dey et al., 1998; Dey et al., 1999).

As definições destes autores estão, de uma determinada forma, interligadas com o contexto do utilizador. Dey apresenta uma definição generalizada para a sensibilidade ao contexto:

“A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task.” (Dey, 2001)³

Segundo estes autores, a ação do utilizador é considerada, nesta definição, como sendo a parte central da sensibilidade ao contexto, pelo que é consistente com a visão defendida nesta tese, sendo por isso a definição a utilizar no nosso trabalho.

Outra característica de relevância na sensibilidade ao contexto são os níveis de sensibilidade da aplicação, que podem ser definidos de várias formas. Na tabela 2, são sumariados diferentes tipos de classificações e, como é possível observar na tabela, o alvo a avaliar em cada classificação é diferente.

³Tradução livre do autor - “O sistema é sensível ao contexto se utilizar o contexto para apresentar informação e/ou serviços relevantes ao utilizador, sendo que a relevância é dependente da ação do utilizador.” (Dey, 2001)

TABELA 2-2 – COMPARAÇÃO DE CLASSIFICAÇÕES DE SENSIBILIDADE AO CONTEXTO

CLASSIFICADOR	SENSIBILIDADE	DESCRIÇÃO
INTERAÇÃO COM O UTILIZADOR (DUSTDAR, BALDAUF, & ROSENBERG, 2007)	PERSONALIZADO	INTERAÇÃO COM O UTILIZADOR PARA DEFINIR PREFERÊNCIAS
	PASSIVO	EXECUÇÃO DE AÇÕES APROPRIADAS SUJEITAS A DECISÃO DO UTILIZADOR
	ATIVO	EXECUÇÃO DE AÇÕES DE FORMA AUTÓNOMA
AQUISIÇÃO SENSORIAL (PERERA ET AL., 2014)	AUTOCONTIDO	A SENSIBILIDADE AO CONTEXTO É OBTIDA POR <i>HARDWARE</i> INDEPENDENTE SEM SUPORTE EXTERNO OU DO EXTERIOR.
	INFRAESTRUTURAL	SISTEMAS EXTERNOS OU INFRAESTRUTURAS PROVIDENCIAM SUPORTE ADICIONAL AOS SISTEMAS SENSÍVEIS AO CONTEXTO
ORIGEM (TUULARI, 2000)	HARDWARE	SENSIBILIDADE AO CONTEXTO É OBTIDA APENAS POR <i>HARDWARE</i>
	SOFTWARE	O CONHECIMENTO É INFERIDO BASEADO EM REPOSITÓRIOS DE CONTEXTO QUE É APLICADO PARA OBTER SENSIBILIDADE CONTEXTUAL
MODELO CONTEXTUAL (LEE, CHANG, & LEE, 2011)	LOCALIZAÇÃO	A SENSIBILIDADE AO CONTEXTO ESTÁ LIMITADA À SENSIBILIDADE DA LOCALIZAÇÃO
	NÍVEL INTERMÉDIO	VÁRIOS TIPOS DE CONTEXTO SÃO USADOS, MAS NÃO É GERIDO INEQUIVOCAMENTE
	SEMÂNTICO	TECNOLOGIAS SEMÂNTICAS SÃO UTILIZADAS PARA MELHORAR A SENSIBILIDADE AO CONTEXTO

2.3.3. Considerações finais

Nesta secção foi apresentada uma revisão da sensibilidade ao contexto, sendo apresentada também a definição proposta por Dey (2001), que será a definição a seguir como base no desenvolvimento desta tese, uma vez que a mesma é consistente com a nossa visão de contexto centrado na ação do utilizador.

2.4. Ciclo de vida do contexto

Nesta secção será apresentada uma revisão, relativa ao ciclo de vida do contexto. Esta revisão será iniciada com a apresentação das diversas etapas pelas quais o contexto irá passar, sendo para cada uma destas etapas, analisados os principais tipos de técnicas mais utilizados. No Capítulo 3 – Revisão do Estado da Arte, será apresentada uma revisão de exemplos de aplicação dessas técnicas.

2.4.1. Introdução

A utilização de contexto varia de acordo com a arquitetura do projeto e com o autor, mas, de uma forma geral, obedece a um ciclo de vida. O ciclo de vida do contexto, inicia-se com a obtenção do contexto e termina com a sua distribuição, após o processamento, para os consumidores interessados. Este ciclo é caracterizado, de uma forma geral, por cinco fases: Aquisição, Modelação, Classificação, Inferência e Distribuição. Estas fases são facilmente reconhecidas nas diversas propostas de *middleware* da literatura (apresentadas no Capítulo 3), incluindo as aplicações sensíveis ao contexto não dependente de serviços terceiros, isto é, aplicações que fazem todo o processo de inferência de contexto apenas para seu uso próprio. A Figura 2-1 apresenta a sequência dessas etapas.



FIGURA 2-1 – CICLO DE VIDA DO CONTEXTO

2.4.2. Aquisição de Contexto

O objetivo da aquisição de contexto é obter o máximo de dados possíveis, de forma a permitir que as aplicações, sensíveis ao contexto, possam tornar-se mais eficientes e eficazes devido à qualidade da informação contextual. Assim, a qualidade de contexto (QoC) apresenta-se como um conceito fundamental na aquisição de contexto (Buchholz & Schiffers, 2003), e nesse sentido, a sua importância é representada por Bellavista, Corradi, Fanelli e Foschini (2012), onde se define qualidade de contexto como um conjunto de parâmetros que expressam a qualidade dos requisitos e as propriedades que definem o atributo contextual.

Os autores resumem que a qualidade de contexto se baseia em 4 variáveis: validade, precisão, atualização e o tempo desde a aquisição do atributo contextual. No entanto, a qualidade do contexto, é dependente da qualidade de diversos fatores, como seja, da precisão e exatidão dos sensores físicos, da qualidade do atributo contextual e da qualidade do processo de distribuição da informação contextual. A qualidade do contexto é importante também para resolver problemas de conflito de atributos contextuais (Bellavista et al., 2012).

Tal como referido anteriormente, existem diversos modos de categorização contextual. No entanto, uma vez que nesta fase do processo se trata do processo de aquisição, apenas é feita uma diferenciação ao nível da origem dos dados e da sua periodicidade, isto é, entre os dados não processados e dados processados e eventos discretos ou contínuos (Ahn & Kim, 2006; Sanchez, Lanza, Olsen, Bauer, & Girod-Genet, 2006).

Contexto não processado – O contexto não processado é obtido diretamente de um conjunto de sensores que são escolhidos consoante as necessidades da aplicação. Neste conceito, um sensor é sensível a apenas um determinado fenómeno e monitoriza as alterações relevantes que, depois, são convertidas num formato passível de ser utilizado. Estes podem ser, desde sensores simples (temperatura) a sensores mais complexos (localização), ou mesmo sensores virtuais.

Contexto processado – O contexto processado refere-se ao contexto que não pode ser obtido diretamente, como por exemplo, conhecimento, preferências, dados históricos, estimativas. Como este tipo de informação depende de diversos fatores, deve ser obtida de outras formas, seja por preenchimento pelo utilizador por processamento de outras informações ou mesmo pela inferência, com base em regras por contextos obtidos anteriormente.

Amostras discretas – dados relativos a um evento contextual que ocorreu num determinado momento no tempo t e que é independente das amostras anteriores e posteriores.

Amostras contínuas – dados relativos a um evento contextual que ocorreu durante um período de tempo, onde cada amostra depende da anterior e posterior no período temporal considerado e as mesmas não podem ser consideradas individualmente.

No entanto, a aquisição varia também da forma como os dados são adquiridos, podendo estes ser obtidos por vários métodos, via acesso direto ou através de *middleware*. Na tabela seguinte é apresentada um resumo das diferentes formas de aquisição de informação contextual.

TABELA 2-3 - COMPARAÇÃO DE MÉTODOS DE AQUISIÇÃO BASEADOS NA ORIGEM

TIPO	VANTAGENS	DESVANTAGENS
ACESSO DIRETO AO SENSOR	<ul style="list-style-type: none"> - COMUNICAÇÃO MAIS EFICIENTE, POIS, POSSIBILITA O ACESSO DIRETO AOS DADOS EM BRUTO; - PERMITE MAIOR CONTROLO SOBRE O PROCESSO DE AQUISIÇÃO E TRANSFORMAÇÃO DOS DADOS; 	<ul style="list-style-type: none"> - É NECESSÁRIO UM CONHECIMENTO ELEVADO SOBRE AS CARACTERÍSTICAS FÍSICAS DO SENSOR O QUE REQUER UM MAIOR INVESTIMENTO AO NÍVEL DE ESFORÇO, TEMPO E FINANCEIRO NECESSÁRIO; - A REUTILIZAÇÃO DE PROCESSOS E MÓDULOS É MAIS REDUZIDA DEVIDO AS CARACTERÍSTICAS ESPECÍFICAS DE CADA SENSOR.
MIDDLEWARE	<ul style="list-style-type: none"> - OBTENÇÃO DE DADOS MAIS RAPIDAMENTE E COM MENOS ESFORÇO TECNOLÓGICO; - MAIOR CAPACIDADE DE SUPORTE A VÁRIOS TIPOS DE SENSORES. 	<ul style="list-style-type: none"> - SÃO NECESSÁRIOS MAIORES RECURSOS COMPUTACIONAIS DO QUE NO ACESSO DIRETO; - A EFICIÊNCIA DE AQUISIÇÃO É MENOR, DEVIDO À NECESSIDADE DE OS DADOS SEREM OBTIDOS VIA CAMADA INTERMÉDIA.

2.4.3. Modelação de contexto

Durante a execução da fase de aquisição de dados sensoriais, os dados são obtidos em grandes quantidades e são tipicamente estruturados consoante o formato disponibilizado pelo sensor. De forma a ser possível utilizar os dados, nas diversas aplicações que deles necessitem, é importante que sejam uniformizados e convertidos num formato adequado para que possam ser armazenados, interpretados e partilhados, entre os vários intervenientes. Existem diversas técnicas que visam dar resposta a esse desafio e, por isso, é comum existirem diversos artigos de revisão que analisam as técnicas mais populares e disseminam as boas práticas já existentes (Strang & Linnhoff, 2004; Moore, Hu, Zhu, Campbell, & Ratcliffe, 2007; Hoareau & Satoh, 2009).

Em seguida nesta secção serão apresentadas as técnicas mais comuns nesta área, obtidas a partir da metodologia descrita na secção de enquadramento deste capítulo, analisando os resultados obtidos pela aplicação das duas primeiras etapas de seleção. As técnicas serão descritas detalhadamente e comparadas no final da subsecção com base nas suas vantagens.

Modelo baseado em Key-Value

Neste modelo são utilizados pares chave-valor para armazenar e organizar os atributos e os valores. Este formato não é restritivo ao nível do formato dos dados a utilizar para os valores. Pela sua simplicidade é um dos mais utilizados, especialmente nas primeiras propostas na área. O modelo baseado em *Key-Value* é utilizado por Schilit et al. (1994) para modelar a informação contextual.

É comum encontrar este modelo em propostas de serviços distribuídos como Buchholz, Hochstatter e Linnhoff-Popien (2007), ou Aiello, Fortino, Gravina e Guerrieri (2011). A principal vantagem deste modelo é a sua simplicidade de uso, mas tem como principal obstáculo a falta de algoritmos para a obtenção da informação de forma eficiente.

Modelo baseado em Markup

Neste modelo é utilizada uma *markup language* para armazenar a informação dentro de *tags* (símbolos ou anotações) que representam e formatam os dados. As principais limitações deste modelo prendem-se com as suas características, as quais obrigam que o modelo hierárquico tenha que ser previamente definido e, principalmente, é muito complicado para ser utilizado para representar relações entre os contextos. Tipicamente, os modelos deste tipo têm origem numa derivação do *Standart Generic Markup Language*, a superclasse de todas as linguagens markup, como o XML. Existem algumas especificações standard como *Composite Capabilities / Preferences Profile (CC/PP)* (*World Wide Web Consortium, 2004*) que visam normalizar a implementação mas, apresentam algumas desvantagens e rigidez.

Alguns autores reuniram as vantagens do *markup*, complementando o CC/PP com as capacidades de uma ontologia RDF. Um exemplo disto é a *Comprehensive Structured Context Profiles (CSCP)* proposto por Held, Buchholz e Schill (2002). No entanto, nem todas se baseiam no CC/PP e, muitas delas ou se apresentam proprietárias ou limitadas a um conjunto restrito de informação contextual, como o *Pervasive Profile Description Language* (Chitchebina & Franz, 2003), ou *Reflective Middleware* (Capra, Emmerich, & Mascolo, 2001), ou o *ECGWARE* (Gonçalves, Filho, & Andreão, 2008).

Modelos Gráficos

Os modelos deste tipo têm a capacidade de representar relações entre os dados. Um dos instrumentos mais utilizados neste formato são os diagramas UML. O UML (*Unified Markup Language*) é um formato standard genérico que possui a capacidade de expressar as relações entre as entidades com uma linguagem gráfica. Em paralelo com o uso de UML, é também comum utilizar-se ERM (*Entity Relationship Model*) e ORM (*Object Role Model*), para o desenho e pesquisa de dados ao nível conceptual.

A principal desvantagem decorre de que a interoperabilidade entre os diferentes sistemas de gestão de bases de dados nem sempre é simples ou, por vezes, possível de forma direta. Um exemplo da aplicação deste modelo é a proposta de Bauer, Kutsche e Ehrmanntraut

(2003), onde a informação contextual do sistema de gestão de tráfego, é modelada como sendo uma extensão UML. Outra proposta deste tipo de modelo é a proposta introduzida por Henricksen, Indulska e Rakotonirainy (2003), onde o contexto é uma extensão do modelo ORM, ou do ContextXML proposto por Sheng e Benatallah (2005), ou da proposta de monitorização remota de Rialle, Lamy, Noury e Bajolle (2003).

Modelos baseados em Orientação aos Objetos

Os modelos deste tipo utilizam conceitos de programação orientada a objetos (encapsulamento, herança, entre outros) podendo, assim, aplicar hierarquias e relações entre as classes, para poder representar os dados do contexto. A sua principal desvantagem é que a taxonomia se pode tornar complexa rapidamente e obriga a que, quem a implemente, a conheça na totalidade.

Algumas das propostas baseadas neste modelo são apresentadas por Beigl, Gellersen e Schmidt (1999), ou o *Active Object Model* proposto em Cheverst, Mitchell e Davies (1999), ou o modelo *Hydrogen* de Hofer et al. (2003), ou o modelo baseado em tecnologia semântica de Zhang, Gu e Wang (2005).

Modelos baseados em Lógica

Nos modelos deste tipo, o contexto é definido como sendo factos, expressões e regras, com um nível muito alto de formalidade. As técnicas deste modelo permitem a verificação de consistência e suportam a capacidade de raciocínio lógico. A principal desvantagem deriva da falta de standards e ferramentas de validação. Uma das primeiras propostas de um modelo baseado em lógica foi avançada por McCarthy e Buvac (1997), onde se introduziu o contexto como uma abstração matemática. Posteriormente, surgiram propostas como Giunchiglia, Maltese e Dutta (2012), que se foca mais na modelação do contexto do que na sua interpretação. Outras propostas nesta categoria são, por exemplo, o *Situation Theory* de Barwise e Perry (1981), o *Extended Situation Theory* de Akman e Surav (1997), e o *Sensed Context Model* de Gray e Salber (2001) e Xu e Cheung (2005).

Modelo Focado no domínio

Este modelo é também conhecido como modelo contextual W4, constituído por um quadruplo (Who, What, Where, When/Quem, O Quê, Onde, Quando). Este modelo foi proposto por Castelli, Rosi, Mamei e Zambonelli (2007), e é muito expressivo e flexível para o uso de armazenamento de dados. Tipicamente, este modelo representa o contexto, dando resposta à seguinte questão: “*Alguém ou algo que realizou alguma atividade num determinado local num determinado momento*”.

Modelo Centrado no Utilizador

Os modelos deste tipo são contruídos com base na perspetiva do utilizador e exploram como o contexto é percecionado pelos utilizadores, ao invés dos dispositivos, serviços ou aplicações. Este modelo, proposto por Hong, Schmidtke e Woo (2007), acrescenta ao modelo focado no domínio (*4W Who, What, Where, When -Quem, O Quê, Onde, Quando*), as variáveis *Como/How* e *Porquê/Why*. Este modelo permite expressar contexto de forma muito organizada, implicando um compromisso entre a complexidade da expressão e a facilidade de uso.

Modelo Baseado em ontologias

Os modelos deste tipo utilizam conceitos, instâncias e relações (os componentes de uma ontologia), permitindo representar o conhecimento de forma formal e compreensiva. Devido às capacidades de representação e partilha do conhecimento, este modelo pode ser utilizado para realizar a ponte entre a interpretação das pessoas e dos sistemas, sendo visto por muitos autores como um dos mais promissores (Strang & Linnhoff, 2004; Moore et al., 2007; Hoareau & Satoh, 2009). O modelo baseado em Ontologias apresenta vantagens em relação aos anteriores ao nível de interoperabilidade, formalidade e reutilização.

Comparação das técnicas de modelação contextual

A Tabela 2-4 resume as diversas propostas de técnicas para modelação contextual, analisando as suas principais aplicações, vantagens e desvantagens.

TABELA 2-4 – COMPARAÇÃO DAS DIFERENTES TÉCNICAS DE MODELAÇÃO CONTEXTUAL

TÉCNICA	APLICAÇÕES	VANTAGENS	DESVANTAGENS
KEY-VALUE	MODELO ADEQUADO PARA UTILIZAÇÃO EM CONTEXTOS LIMITADOS E EM APLICAÇÕES QUE SEJAM INDEPENDENTES E NÃO PARTILHEM DADOS.	SIMPLICIDADE E FACILIDADE DE USO; FLEXÍVEL	FALTA DE NORMALIZAÇÃO, E FERRAMENTAS DE VALIDAÇÃO. INÚTIL EM GRANDES VOLUMES DE DADOS;
MARKUP	MODELO ADEQUADO PARA TRANSFERÊNCIA DE DADOS POR REDE. ADEQUADO PARA APLICAÇÕES COM BAIXO NÍVEL DE INFORMAÇÃO.	ESTRUTURADO, E COM ALGUMAS FERRAMENTAS DE VALIDAÇÃO DISPONÍVEIS. FLEXÍVEL DE USO E ELEVADA INTEROPERABILIDADE.	FALTA DE NORMALIZAÇÃO. PROBLEMAS DE REPRESENTAÇÃO DE RELAÇÕES, DEPENDÊNCIAS, E VALIDAÇÃO DE INCONSISTÊNCIA.
GRÁFICO	MODELO ADEQUADO PARA OBTER UM MODELO ENTIDADE-RELAÇÃO QUE É ÚTIL PARA CONTRUIR UMA BASE DE DADOS RELACIONAL.	FORTE CAPACIDADE DE EXPRESSÃO, E COM SUPORTE A RELAÇÕES ENTRE OS DADOS. A VALIDAÇÃO É POSSÍVEL ATRAVÉS DO USO DE RESTRIÇÕES. VÁRIOS STANDARDS DISPONÍVEIS.	PROBLEMAS AO NÍVEL DA INTEROPERABILIDADE.
ORIENTADO AOS OBJETOS	MODELO ADEQUADO PARA SER USADO EM APLICAÇÕES COM GRANDES CAPACIDADES COMPUTACIONAIS QUE UTILIZEM LÍNGUAS DE PROGRAMAÇÃO DE ALTO NÍVEL.	AS RELAÇÕES SÃO PERMITIDAS. EXISTÊNCIA DE ALGUMAS FERRAMENTAS DISPONÍVEIS. PODE SER FACILMENTE MANIPULADO COM AS LINGUAGENS DE PROGRAMAÇÃO EXISTENTES.	FALTA DE NORMALIZAÇÃO, E FERRAMENTAS DE VALIDAÇÃO. DIFÍCIL DE OBTER INFORMAÇÃO.
LÓGICO	MODELO ADEQUADO PARA APLICAÇÕES EM QUE É NECESSÁRIO UM ALTO NÍVEL INFORMATIVO, E QUE OS PROGRAMADORES CONSIGAM DEFINIR TODAS AS RESTRIÇÕES.	GRANDE CAPACIDADE DE EXPRESSÃO. SUPORTE AO RACIOCÍNIO LÓGICO. VERIFICAÇÃO DE CONSISTÊNCIA. FERRAMENTAS DE PROCESSAMENTO DISPONÍVEIS.	FALTA DE NORMALIZAÇÃO, E FERRAMENTAS DE VALIDAÇÃO.
CENTRADO NO DOMÍNIO	MODELO ADEQUADO PARA SER USADO EM APLICAÇÕES FOCADAS APENAS NO USO EM DOMÍNIO ÚNICO.	EXPRESSIVO, FLEXÍVEL E ESTRUTURADO. FACILIDADE DE COMPREENSÃO.	FALTA DE NORMALIZAÇÃO, E FERRAMENTAS DE VALIDAÇÃO.

CENTRADO NO UTILIZADOR	MODELO ADEQUADO PARA SER USADO EM APLICAÇÕES FOCADAS NA PERSPETIVA DO UTILIZADOR, COM REPRESENTAÇÃO DE FORMA INTUITIVA.	O CONTEXTO É EXPRESSADO DE FORMA ORGANIZADA. ESCALÁVEL COMO CRESCIMENTO DOS DADOS. PERMITE RACIOCÍNIOS LÓGICOS.	FALTA DE NORMALIZAÇÃO, E FERRAMENTAS DE VALIDAÇÃO. COMPLEXO DE USAR. FALTE DE CAPACIDADE DE FORMALIZAÇÃO.
ONTOLÓGICO	MODELO ADEQUADO PARA UTILIZAÇÃO EM APLICAÇÕES QUE SEJA NECESSÁRIO PARTILHAR A INFORMAÇÃO COM TERCEIROS.	SUPORTA RACIOCÍNIOS LÓGICOS. FORTE EXPRESSIVIDADE. PERMITE RELAÇÕES. FORTE CAPACIDADE DE VALIDAÇÃO, STANDARDS ESTÁVEIS E DE QUALIDADE.	A REPRESENTAÇÃO PODE TORNAR-SE COMPLEXA. A OBTENÇÃO DOS DADOS PODE SER COMPLEXA. INCAPACIDADE DE RESPONDER À INCERTEZA.

De uma forma geral é possível verificar, pela observação da Tabela 2-4, que nenhuma das metodologias é suficiente por si só. Todas elas possuem limitações e vantagens sobre as outras, consoante a situação em si.

Comparativamente, o modelo baseado em ontologias é o mais promissor, por resolver muitos dos principais desafios tais como, interoperabilidade, raciocínios lógicos, forte validação, formalização e elevada capacidade de expressão. A ontologia, na sua vertente clássica, não dá resposta a situações de incerteza, imprecisão e falta de exatidão no conhecimento, o que é uma situação muito comum nas aplicações reais (Lukasiewicz & Straccia, 2008). Para colmatar esta falha, surgiram algumas propostas híbridas, como a de Liang e Cao (2015), que integra os conceitos do modelo gráfico com o modelo ontológico. Outra proposta que se destaca pela sua diferença, é a proposta de Bobillo e Straccia (2011) que reúne as vantagens dos conceitos da lógica difusa no modelo ontológico.

2.4.4. Inferência de contexto

A inferência de contexto surge devido à natureza imperfeita e incerta dos dados contextuais. O objetivo da inferência é deduzir informação contextual, de alto nível, a partir de

dados em bruto. A inferência permite também algumas funcionalidades base tais como, validação dos dados, preenchimento de dados em falta, detecção e remoção de anomalias, verificação da consistência, e obtenção de novos dados através de cálculo a partir dos existentes (Ameyed, Miraoui, & Tadj, 2011). De forma a permitir ter uma noção do tipo de técnicas de inferência existentes iremos, em seguida, apresentar um resumo das técnicas mais populares obtidas a partir da metodologia descrita na secção de enquadramento deste capítulo, analisando os resultados obtidos pela aplicação das duas primeiras etapas de seleção. As técnicas serão descritas detalhadamente apresentadas por ordem alfabética e comparadas no final da subsecção com base nas suas vantagens.

Bayesian Network

Este modelo probabilístico gráfico representa um conjunto aleatório de variáveis e as suas dependências condicionais através de um grafo (Neapolitan, 1990). Uma rede deste tipo permite representar a probabilidade de relações entre variáveis e resultados. Assim, dado um conjunto de variáveis, esta rede consegue calcular a probabilidade de ocorrência de vários tipos de resultados. A principal desvantagem desta técnica é que não é possível definir modelos cíclicos. No entanto, esta desvantagem pode ser ultrapassada com o uso de *Dynamic Bayesian Network*, que é um tipo especial de rede, que permite relacionar as variáveis entre si, através do uso de passos paralelos.

Context Space Theory

Esta proposta foi desenvolvida com vista a definir a sensibilidade ao contexto e lidar com os problemas sensoriais que criam incerteza (Lenat, 1998). Esta teoria utiliza metáforas espaciais para representar o contexto num espaço multidimensional, onde o número de dimensões é equivalente ao número de atributos contextuais no estado contextual. Foi construída com o objetivo de tornar a sensibilidade ao contexto simples.

Nesta teoria, qualquer tipo de dados que sejam usados para inferir sobre o contexto, são designados por atributos contextuais. Estes atributos são medidos diretamente dos sensores, ou

calculados a partir de outros atributos contextuais. Um estado contextual é constituído por todo o contexto relevante num determinado ponto no tempo. O conjunto de todos os estados contextuais definem o espaço aplicacional.

Data-Mining

Os algoritmos de *data-mining* ou de descoberta de conhecimento são processos computacionais que tentam prever padrões em grandes volumes de dados (Hastie, Tibshirani, & Friedman, 2001). O objetivo destes algoritmos é extrair informação a partir de um conjunto de dados e transformá-lo numa estrutura capaz de ser utilizada posteriormente. Os algoritmos deste tipo são classificados tipicamente em 6 tipos:

- **Deteção de anomalias** – deteção de casos que não correspondem aos valores médios típicos, ou considerados fora do comum;
- **Aprendizagem de associações** – procura por relações entre as variáveis;
- **Aglomeracão** – descobrir estruturas nos dados que são semelhantes sem que se conheçam as estruturas previamente;
- **Classificação** – generalizar as estruturas para aplicar a dados novos;
- **Regressão** – tentar modelar uma função que representa a evolução dos dados com o menor erro.
- **Sumarização** – apresentar, sob várias formas de visualização e/ou relatórios, uma representação mais compacta dos dados analisados.

A principal desvantagem desta abordagem para o uso na inferência de contexto é que implica a existência inicial de um elevado conjunto de dados para serem analisados e que a qualidade dos resultados está relacionada com a quantidade e qualidade dos dados utilizados.

Domínio temporal

As técnicas em domínio temporal são as mais populares no processamento em tempo real, uma vez que as amostras são capturadas em sequência temporal (Lee, Cheatham, & Wiesner, 1950). Algumas das suas principais vantagens estão associadas à capacidade de assumir que os sinais são deterministas e limitados temporalmente, permitindo assim excluir anomalias no sinal. Alguns exemplos de técnicas utilizadas para manipular amostras temporais são apresentados na Tabela 2-5.

TABELA 2-5 – RESUMO DE VÁRIOS PARAMETROS DE PROCESSAMENTO EM DOMÍNIO TEMPORAL

TIPO	DESCRIÇÃO
VARIÂNCIA	REPRESENTA A DINÂMICA DA ATIVIDADE DA AMOSTRA.
DESVIO PADRÃO	REPRESENTA A DINÂMICA DA AMOSTRA, INCLUINDO O QUANTO OS VALORES DA AMOSTRA ESTÃO DISTRIBUÍDOS.
RAIZ QUADRÁTICA MÉDIA	REPRESENTA A INTENSIDADE DA AMOSTRA
CORRELAÇÃO ENTRE AMOSTRAS	PERMITE DISTINGUIR ATIVIDADES SIMILARES QUE OCORREM EM UMA OU MÚLTIPLAS DIMENSÕES.
PICO ESPECTRAL	REPRESENTA A FREQUÊNCIA DOMINANTE DO SINAL
INTERVALOS INTERQUARTIS	REPRESENTA A DISPERSÃO DO SINAL TENDO EM CONTA OS VALORES EXTREMOS DE DUAS AMOSTRAS
DESVIO MÉDIO ABSOLUTO	REPRESENTA A MÉDIA DA DISPERSÃO DO SINAL EM RELAÇÃO AO SEU VALOR MÉDIO.
CENTROIDS	REPRESENTA O VALOR CENTRAL MAIS EQUILIBRADO NA DISTRIBUIÇÃO DO SINAL.
HISTOGRAMA	REPRESENTA A DISTRIBUIÇÃO DA DENSIDADE DO SINAL.
KURTOSIS	REPRESENTA A DIMENSÃO DO PICO DA DENSIDADE DO SINAL.

Domínio em frequência

As técnicas de domínio em frequência são técnicas semelhantes às anteriores, mas baseando-se na variação da frequência do sinal, no entanto, estas requerem mais tempo de computação para discriminar valores como a periodicidade de uma amostra (Boashash, 1992). Assim, para a deteção de um pico em frequência, é registado um índice da frequência com o

maior nível de energia. Este tipo de técnica é utilizado em situações em que o contexto é obtido através de uma amostra periódica como, por exemplo, o valor obtido através de um acelerómetro, para o cálculo do movimento de caminhar, ou correr (Mäntyjärvi, Himberg, Korpipää, & Mannila, 2001)

Expert System

Estas técnicas são ferramentas capazes de reproduzir os processos cognitivos de um especialista num determinado domínio (Jackson, 1998). Eles são desenhados para resolver problemas de elevada complexidade, raciocinando sobre o conhecimento, representado principalmente através de regras *se* → *então*.

Estes sistemas dividem-se em dois subsistemas: (1) o motor de inferência, e (2) a base de dados de conhecimento. A base de dados representa os factos e as regras. O motor de inferência aplica as regras aos factos conhecidos e deduz novos factos, tomando assim decisões. O objetivo final deste sistema é definir regras de previsão. Esta metodologia permite uma visão clara de todo o sistema.

Markov Chains

Esta abordagem caracteriza-se por um conjunto de estados, o processo inicia-se num estado e vai avançando, progressivamente, de um estado para outro (Norris, 1998). Cada movimento é designado por step (passo). Segundo a teoria, se a cadeia se encontra no estado X_i , então irá progredir para o estado X_j no próximo passo com a probabilidade p_{ij} , e esta probabilidade não depende dos estados anteriores. $[X_n, n > 0]$ X_n é o estado a ser processado no tempo n , $p_{ij}(n)$, então designado como a probabilidade de transição, do estado i para o estado j , no tempo n .

Uma das principais desvantagens deste modelo é a limitação de números de estados sobre uma linha de tempo discreta, no entanto, este modelo continua a ser muito viável para a maioria dos problemas de inferência contextual. Para situações mais complexas, é possível utilizar algumas das extensões a este modelo como por exemplo a *hidden markov chain*.

Neural Networks

Estes são modelos inspirados no funcionamento do nosso cérebro, mais propriamente, no funcionamento dos neurónios (Haykin, 1998). Estas são uma formalização matemática que tenta emular a estrutura dos neurónios biológicos. Esta abordagem tornou-se rapidamente uma das mais populares para resolver problemas pertinentes de inteligência artificial, nomeadamente, reconhecimento de padrões, associação de modelos, previsões, etc. Existem vários tipos de redes neuronais e as abordagens de inferência de contexto baseadas nesta técnica, são muito flexíveis e eficientes.

A principal desvantagem das redes neuronais decorre de funcionarem como caixas negras, em que não é possível analisar o modo de funcionamento interno. Outras desvantagens passam por não possuírem tolerância para ser usada para o *data-mining* e terem uma fase de treino significativamente lenta.

Sequence Prediction Approach

Nesta abordagem o problema é descrito como sendo uma sequência de eventos ou estados recebidos num intervalo de tempo de 1 até t (Sun & Giles, 2001). O objetivo é reduzi-los, prevendo o próximo conjunto de eventos ou o próximo estado. Apesar de bastante eficaz, esta abordagem, apresenta algumas desvantagens, tais como, só poder ser usada quando o contexto é descrito como um fluxo de eventos ou a incapacidade de lidar com o tempo e, portanto, não conseguir detetar padrões temporalmente dependentes.

Similarity and Semantic Similarity

Esta abordagem utiliza as relações existentes entre os valores contextuais de forma a deduzir o contexto emergente (Feng, Bagheri, Ensan, & Jovanovic, 2017). As relações entre o contexto podem ser de 2 tipos: singulares ou múltiplas. As relações entre os valores contextuais identificam entre 2 valores contextuais de ontologias diferentes baseados na semelhança entre eles.

A semelhança entre estes valores contextuais pode ser, desde igualdade entre propriedades atômicas, a mesma variável com nome e valor igual em ambos, até igualdade entre duas classes ou valores não atômicos. Com base na existência desta relação podem ser identificadas quatro tipos de relações: (1) interseção, (2) complementação, (3) equivalência, (4) independência.

Comparação das técnicas de inferência contextual

A Tabela 2-6 resume as diversas propostas de técnicas para modelação contextual, analisando as suas principais aplicações, vantagens e desvantagens.

TABELA 2-6 – COMPARAÇÃO DAS DIFERENTES TÉCNICAS DE INFERÊNCIA CONTEXTUAL

TÉCNICA	APLICAÇÕES	VANTAGENS	DESvantagens
MARKOV MODELS / SEQUENCE PREDICTION APPROACH	PARA SITUAÇÕES ONDE AS PROBABILIDADES DE OCORRÊNCIA SÃO CONHECIDAS E A COMBINAÇÃO DE VÁRIAS FONTES É NECESSÁRIA.	PERMITE COMBINAR DADOS DE VÁRIOS SENSORES; PERMITE DAR RESPOSTA A SITUAÇÕES NÃO PREVISTAS, DANDO RESPOSTA A INCERTEZA.	É NECESSÁRIO SABER ANTECIPADAMENTE AS PROBABILIDADES DE CADA SITUAÇÃO.
EXPERT SYSTEM / SIMILARITY AND SEMANTIC SIMILARITY	UTILIZADO EM SITUAÇÕES EM QUE OS DADOS NECESSITEM DE SER CONVERTIDOS PARA INFORMAÇÃO DE ALTO NÍVEL. ÚTEIS PARA DEFINIR EVENTOS E SITUAÇÕES CONHECIDAS.	SIMPLES DE DEFINIR, E COM ELEVADA CAPACIDADE DE EXPANSÃO; MENOR NÍVEL DE RECURSOS NECESSÁRIOS;	É NECESSÁRIO DEFINIR AS REGRAS MANUALMENTE; SEM VALIDAÇÃO OU VERIFICAÇÃO DE QUALIDADE; GRANDE DIFICULDADE EM LIDAR COM A INCERTEZA E SITUAÇÕES INESPERADAS.
CONTEXT SPACE THEORY	PARA SITUAÇÕES EM QUE O CONHECIMENTO CONTEXTUAL É FUNDAMENTAL PARA A APLICAÇÃO DE RACIOCÍNIOS SOBRE OS DADOS.	PERMITE RACIOCÍNIOS E REPRESENTAÇÕES COMPLEXAS; PERMITE OBTER RESULTADOS MAIS SIGNIFICATIVOS CONTEXTUALMENTE	NECESSÁRIA MODELAÇÃO DE DADOS PARA UM FORMATO COMPATÍVEL; RACIOCÍNIOS LIMITADOS; PERFORMANCE MAIS REDUZIDA.

BAYESIAN NETWORK /NEURAL NETWORKS /DATA MINING	PARA SITUAÇÕES EM QUE O RESULTADO É CONHECIDO, E EXISTEM ELEVADAS QUANTIDADES DE DADOS DISPONÍVEIS (TANTO PARA TREINO COMO AVALIAÇÃO).	PRECISÃO E EXATIDÃO ELEVADA; ELEVADO NÚMERO DE MODELOS ALTERNATIVOS; MODELO DE FUNCIONAMENTO ESTATÍSTICO.	REQUER UMA QUANTIDADE DE DADOS SIGNIFICATIVO; PODE SER COMPUTACIONALMENTE INTENSIVO O SEU PROCESSAMENTO; REQUER TREINO ANTECIPADO DO MODELO.
DOMÍNIO TEMPORAL /FREQUÊNCIA	PARA SITUAÇÕES EM QUE OS RESULTADOS NÃO SÃO CONHECIDOS,	NÃO É NECESSÁRIO TREINO PRÉVIO DOS MODELOS; NÃO É NECESSÁRIO SABER QUAL O RESULTADO POSSÍVEL PREVIAMENTE.	MODELOS PODEM TORNAR-SE COMPLEXOS; DIFÍCIL DE VALIDAR E COM RESULTADOS NÃO PREVISÍVEIS. PODEM SER COMPUTACIONALMENTE INTENSIVOS.

2.4.5. Considerações Finais

Nesta secção foi apresentado o ciclo de vida do atributo e modelo contextual, desde a sua origem, ex. por leitura de sensores, até ao seu processamento por processos mais complexos e algoritmos de inferência. Foram apresentadas, também, as principais técnicas de modelação e inferência. Apresentou-se um comparativo com as vantagens e desvantagens de cada um dos modelos e técnicas, onde ficou claro que cada técnica tem as suas particularidades e que é necessário escolher a mais adequada, de acordo com o problema a resolver.

2.5. Desafios das técnicas de inferência contextual

Nem todo o tipo de contexto pode ser obtido diretamente a partir de sensores externos e mesmo, por vezes, estes não estão disponíveis pelo que é necessário utilizar outras técnicas para conseguir inferir o contexto do utilizador. Assim, nesta secção será apresentada uma revisão de algumas técnicas de inferência contextual que visam tentar dar resposta à necessidade de inferência contextual utilizadas no caso de estudo prático desta tese. De acordo com o ciclo de vida apresentado na secção anterior, estas técnicas são incluídas na fase de inferência de contexto.

2.5.1. Localização interna

O contexto localização é um dos contextos mais utilizados nas propostas de sensibilidade ao contexto pois, através dele, é possível obter otimizar um conjunto de informação que é mais relevante ao utilizador do que seria ao apresentar uma listagem não otimizada.

O contexto localização varia ao nível da granularidade que se pretende, podendo variar de zona, (país, região ou cidade) ou mais detalhado (bairro, rua ou posição específica GPS), ou até mesmo dentro de edifícios em qual das salas ou lojas se encontra. Relativamente às primeiras categorias de localização a problemática é facilmente resolvida com a utilização do sistema GPS, mas quando se pretende obter a informação relativa ao interior de edifícios, a situação fica muito mais complicada, uma vez que devido às características do sinal GPS, este não se propaga muito bem no interior destes.

Devido ao interesse nesta problemática, existem várias possibilidades de realizar a localização interna, com algumas propostas que visam localizar um utilizador invisível utilizando a tecnologia RFID (Fernandes, Faria, Martins, Paredes, & Barroso, 2013; Fernandes,

Filipe, Costa, & Barroso, 2014). No entanto, propostas que necessitem de equipamentos externos não são muito interessantes para dar resposta ao nosso desafio, uma vez que aumenta a complexidade de instalação dos mesmos, e a granularidade que se pretende obter não necessita de uma precisão tão elevada, uma vez que o que principalmente se pretende é saber a sala onde este se encontra, e não o qual a cadeira específica.

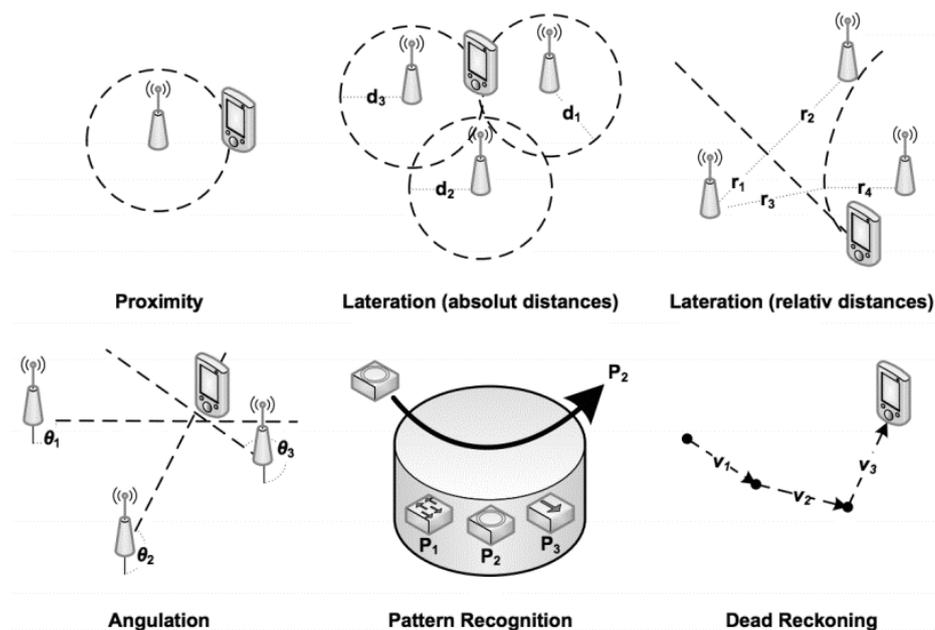


FIGURA 2-2 – TÉCNICAS DE LOCALIZAÇÃO INDOOR (ADAPTADO DE KJÆRGAARD, 2007)

Como é possível verificar na Figura 2-2, existem várias técnicas que visam a localização interna, que começam pelas mais simples, utilizando a proximidade a um determinado transmissor, à tri-lateração e angulação que utiliza 3 dispositivos e as distâncias ou os ângulos entre eles e os transmissores. Existem, por fim, técnicas mais avançadas como o *Dead Reckoning*, que a partir de um ponto inicial conhecido e tem em conta as deslocções realizadas para o cálculo da posição atual, e por fim, através do reconhecimento de padrões, que utilizando técnicas de aprendizagem artificial, que tentam calcular a posição com base em mapeamentos anteriores. É neste ultimo tipo que se irá focar a nossa revisão devido à simplicidade de implementação e reconhecimento de novas zonas.

Este conceito funciona baseando-se na impressão digital (*BSSID*) que cada dispositivo possui e na força de sinal (*RSS - Received Signal Strength*) que o dispositivo recebe desse

equipamento, que pode ser correlacionada com a distância ao mesmo. Esta informação é recolhida pelo equipamento móvel sempre que ele realiza um scan de redes Wi-Fi disponíveis, e constrói uma lista com a força de sinal em cada local. A Figura 2-3, representa uma visualização desse conceito, onde é possível verificar que a força de sinal varia com a proximidade a cada um dos emissores WiFi e, por isso, é possível assim inferir qual a divisão onde o mesmo de encontra.



FIGURA 2-3 – LOCALIZAÇÃO BASEADA EM IMPRESSÃO DIGITAL POR WIFI (FIND PROJECT, 2016).

No entanto, este problema de categorização é um pouco mais complexo, pois existem diversos problemas que prejudicam o sinal e, nem sempre, se consegue realizar um mapeamento direto entre a força de sinal e a sala onde se encontra. Por esse motivo é necessário aplicar técnicas de que permitam a melhoria contínua do algoritmo aplicado. Sendo um problema de categorização, uma abordagem que tem apresentado resultados muito interessantes é a localização em interiores utilizado classificadores baseados em árvores como o *Random Forest* (Breiman, 2001), que apresenta uma boa solução para o problema, principalmente ao nível da precisão e robustez de funcionamento e melhoramento contínuos (Calderoni, Ferrara, Franco, & Maio, 2015; Górak & Luckner, 2016).

A técnica *Random Forest*, é um classificador que consiste numa coleção de classificadores hierárquicos do tipo árvore $\{h(\mathbf{x}, \theta_k), k = 1, \dots\}$ onde o $\{\theta_k\}$ é um vetor independente dos vetores anteriores $\{\theta_1, \dots, \theta_{k-1}\}$, mas com a mesma distribuição. Assim, uma árvore é construída a partir do conjunto de treino e θ_k , resultando num classificador $h(\mathbf{x}, \theta_k)$, onde \mathbf{x} , é o vetor de entrada. Após a construção de um largo conjunto de árvores, elas irão votar na classe mais popular em obtida de cada árvore a partir da entrada \mathbf{x} (Breiman, 2001).

Assim, dado o conjunto de classificadores, $h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_k(\mathbf{x})$, com o conjunto de treino obtido aleatoriamente a partir da distribuição do vetor aleatório Y, \mathbf{X} , permite definir a função de margem que mede a média do número de votos obtidos em \mathbf{X}, Y , para a classe que excede a votação média de qualquer outra classe. Assim, quanto maior a margem, maior a confiança. A função de margem para a *Random Forest* é então:

$$mr(\mathbf{X}, Y) = P_\theta(h(\mathbf{X}, \theta) = Y) - \max_{j \neq Y} P_\theta(h(\mathbf{X}, \theta) = j)$$

onde a força de cada conjunto de classificadores $\{h(\mathbf{X}, \theta)\}$ é

$$s = E_{\mathbf{X}, Y} mr(\mathbf{X}, Y)$$

Assumindo que $s \geq 0$, erro de generalização, que nos permite verificar se a probabilidade é superior ao espaço \mathbf{X}, Y é

$$PE^* \leq var(mr)/s^2$$

No entanto é possível expandir a variância de mr derivando-a da seguinte forma: onde

$$\hat{j}(\mathbf{X}, Y) = \arg \max_{j \neq Y} P_\theta(h(\mathbf{X}, \theta) = j)$$

então,

$$\begin{aligned} mr(\mathbf{X}, Y) &= P_\theta(h(\mathbf{X}, \theta) = Y) - P_\theta(h(\mathbf{X}, \theta) = \hat{j}(\mathbf{X}, Y)) \\ &= E_\theta[(I(\mathbf{X}, \theta) = Y) - I_\theta(h(\mathbf{X}, \theta) = \hat{j}(\mathbf{X}, Y))]. \end{aligned}$$

No entanto, existem algumas considerações a tomar relativamente a aplicação de *Random Forests*, quando aplicada a mais que 2 classes, nomeadamente relativo à função s uma vez que está dependente não só da floresta em si, mas também das árvores individuais, uma vez que é a floresta que determina $\hat{j}(\mathbf{X}, Y)$. Assim

$$PE^* = P_{\mathbf{X}, Y} \left(P_{\theta}(h(\mathbf{X}, \theta) = Y) - \max_{j \neq Y} P_{\theta}(h(\mathbf{X}, \theta) = j) < 0 \right)$$

$$\leq \sum_j P_{\mathbf{X}, Y} (P_{\theta}(h(\mathbf{X}, \theta) = Y) - P_{\theta}(h(\mathbf{X}, \theta) = j) < 0)$$

define

$$s_j = E_{\mathbf{X}, Y} (P_{\theta}(h(\mathbf{X}, \theta) = Y) - P_{\theta}(h(\mathbf{X}, \theta) = j) < 0)$$

como sendo a força de cada conjunto de classificadores $\{h(\mathbf{x}, \theta_k)\}$ em relação à classe j .

Modelo de localização

Com base nesta implementação, Górak e Luckner (2016), propõe o seguinte modelo de localização, onde define o seguinte conjunto de pressupostos:

- (i) AP é o conjunto de Pontos de acesso utilizados para a localização;
- (ii) $F = \mathbf{R}^2 \times \mathbf{Z} \times \mathbf{R} \times \mathbf{R}^n$ é o espaço de todas as possíveis amostras (fingerprints - f), onde $n = \#AP$, onde para $f \in F$:
 - a. As primeiras coordenadas $f_1[m]$, $f_2[m]$ representam a posição horizontal onde a amostra foi recolhida, $f_3 \in \mathbf{Z}$, é o número do andar;
 - b. $f_4[s]$ é o registo temporal da amostra;
 - c. $f_k[\text{dBm}]$, onde $4 \leq k \leq n + 4$, é o RSS (potencia de sinal) da amostra k do AP.

- (iii) $L = (L_x, L_y, L_f) : F \rightarrow R^2 \times Z$ é a projeção para o modelo de 3 coordenadas onde o conjunto F e $\pi : F \rightarrow R^n$ é a projeção para as últimas n coordenadas. Assim L_f representa a associação entre a localização de uma amostra f , e $\pi(f)$ é o vector de RSS (sinal) associado com essa amostra.

Assim, o autor propõe o modelo de localização como a função $\hat{L}: R^n \rightarrow R^2 \times Z$ que dado um registo de sinal $v \in R^n$, $\hat{L}(v)$ prevê a localização onde essa amostra v foi recolhida. Assim para cada $a \in AP$, é construído \hat{L}_x^a , \hat{L}_y^a e \hat{L}_f^a com $\hat{L}(\pi(v)) = (\hat{x}, \hat{y}, \hat{f})$

$$\hat{x} = \text{mean}\{\hat{L}_x^a(v) : a \in \text{supp}(v)\}$$

$$\hat{y} = \text{mean}\{\hat{L}_y^a(v) : a \in \text{supp}(v)\}$$

$$\hat{f} = \text{mean}\{\hat{L}_f^a(v) : a \in \text{supp}(v)\}$$

O exemplo aqui apresentado demonstra a implementação da técnica *Random Forest* para a construção de um modelo de localização que permite a localização interna dentro de edifícios com uma precisão significativa, com Górak e Luckner (2016) a obter taxas de erro perto dos 4,1m horizontal e 4,7 vertical no percentil 90, o que é bastante significativo dada as condições de detioração que o sinal WiFi sofre com os obstáculos.

2.5.2. Reconhecimento de atividade humana

A identificação da atividade que o utilizador está a realizar é em diversas situações de fulcral importância, pois é a partir da existência ou inexistência de atividade ou movimento que se pode, de forma eficiente, otimizar um conjunto de eventos, nomeadamente conseguir prever algumas necessidades. Devido ao evoluir dos equipamentos móveis, muitos destes já possuem um conjunto sofisticado de sensores, como acelerómetros, giroscópios, entre muito outros.

Estes dispositivos abrem as portas à capacidade de monitorizar e recolher dados para vários fins, como o reconhecimento de atividade humana (*Human Activity Recognition - HAR*).

Existem várias propostas na literatura de como é possível inferir a atividade humana a partir de um conjunto de dados (e.g. Manzoor et al., 2010; Pl, #246, tz, Hammerla, & Olivier, 2011; Anguita, Ghio, Oneto, Parra, & Reyes-Ortiz, 2012; Gordon, Czerny, & Beigl, 2014), no entanto a eficácia dessas abordagens depende principalmente dos métodos de extração de características, uma vez que a extração manual destas requer um conhecimento profundo do domínio.

Uma alternativa a este processo é a aplicação de redes neurais, especialmente técnicas baseadas em *Deep Learning* que são aplicadas com bastante sucesso para resolver problemas de reconhecimento como objetos, traduções entre outros. Também nesta abordagem existe algum trabalho, nomeadamente Yang, Nguyen, San, Li e Krishnaswamy (2015), onde utiliza redes neurais convolutivas (CNN) para identificar automaticamente as características da atividade humana ao longo do dia, a partir da informação do acelerómetro.

As CNN, apresentam grande potencial para identificar os padrões salientes dos sinais das HAR, mais especificamente nas unidades de processamento de baixo nível, obtêm a variância do sinal que caracteriza a natureza de cada movimento básico na atividade humana. Assim, os desafios neste problema incluem o processamento das unidades na CNN o qual necessita de ser aplicados na dimensão temporal e a partilha e a unificação dessas unidades entre diversos sensores. Assim, os autores propõem a seguinte arquitetura (Figura 2-4) a utilizar na HAR, onde as letras *c,s,u,o* correspondem às operações *convolution*, *subsampling*, *unification* e *output* respetivamente.

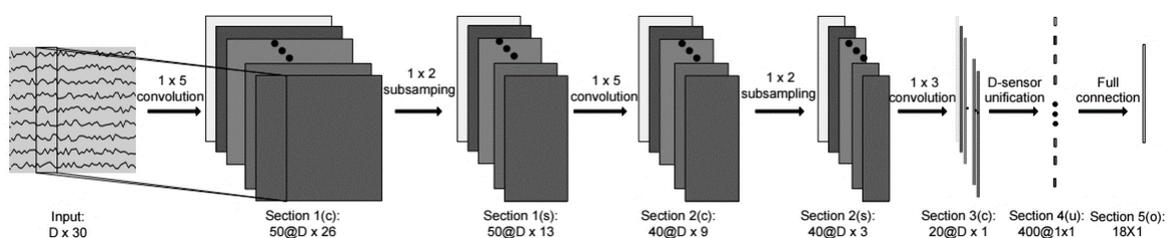


FIGURA 2-4 – ILUSTRAÇÃO DE UMA ARQUITETURA MULTISSENSORIAL UTILIZADA PARA O RECONHECIMENTO DE ATIVIDADE HUMANA (YANG ET AL., 2015)

Este processo é iniciado com uma estratégia de *sliding-window* para obter uma amostra de sinal no tempo. Mais especificamente a transformação numa matriz bidimensional contendo r amostras, cada uma com D atributos. Assim, r é escolhido como sendo a taxa de amostragem, e a amostra j , na camada i da CNN, é uma matriz e o valor na linha x para o sensor d , será representado como sendo $v_{ij}^{x,d}$

Nas camadas de convolução, os mapas de características são convoluídos com *kernels* convolucionais, de forma a permitir serem aprendidos na fase de treino. Formalmente o valor de $v_{ij}^{x,d}$, é obtido a partir de:

$$v_{ij}^{x,d} = \tanh \left(b_{ij} + \sum_m \sum_{p=0}^{P_i-1} w_{ijm}^p v_{(i-1)m}^{x+p,d} \right), \quad \forall d = 1, \dots, D$$

Onde $\tanh(\cdot)$ é a função hiperbólica da tangente e b_{ij} é a tendência para o mapa de características, m indexa sobre os mapas das camadas $(i - 1)$ da característica atual, e w_{ijm}^p é o valor na posição p do *kernel* convolucional, P_i é o comprimento desse *kernel*. Nas camadas de agrupamento, as resoluções dos mapas de características são reduzidas para aumentar a invariância das mesmas à distorção na entrada do sinal. Mais especificamente, os mapas de características das camadas anteriores são agrupados sobre uma vizinhança temporal através de uma função maximizante:

$$v_{ij}^{x,d} = \max_{1 \leq q \leq Q_i} \left(v_{(i-1)j}^{x+p,d} \right), \quad \forall d = 1, \dots, D$$

ou função agrupamento por soma

$$v_{ij}^{x,d} = \frac{1}{Q_i} \sum_{1 \leq q \leq Q_i} \left(v_{(i-1)j}^{x+p,d} \right), \quad \forall d = 1, \dots, D$$

onde Q_i é o comprimento da região de agrupamento. Assim, de forma simplificada, a CNN agrupa-se em 5 secções, onde as duas primeiras secções são constituídas por: *i*) uma camada convolucional que convoluciona a entrada ou a camada anterior com o *kernel* a aprender, *ii*) a

unidade de retificação que realiza o mapeamento dos outputs da camada anterior pela função $relu(v) = \max(v, 0)$; *iii*) uma camada maximizante que localiza o mapa de características máximo numa vizinhança temporal (que normalmente inclui uma operação de subamostragem); *iv*) uma camada de normalização que normaliza os valores de diferentes mapas de características das camadas anteriores:

$$v_{ij} = v_{(i-1)j} \left(k + \alpha \sum_{t \in G(j)} v_{(i-1)t}^2 \right)^{-\beta}$$

onde, k, α, β são hiper parâmetros e $G(j)$ é o conjunto dos mapas de características envolvidos na normalização.

A terceira secção, é constituída apenas por 3 camadas, convolucional, retificação e normalização, uma vez que as saídas desta secção $D \times 1$, e por isso a camada de agregação não é recomendada.

Já na quarta secção, pretende-se unificar todos os mapas de características da terceira secção em todos os sensores D . Para realizar este objetivo, nesta camada existe uma função de concatenação-paramétrica, onde o mapa de características j é calculado por

$$v_{ij} = \tanh \left(b_{ij} + \sum_m \sum_{d=1}^D w_{ijm}^d v_{(i-1)m}^d \right),$$

e após esta unificação seguem novas camadas de retificação e normalização.

A quinta secção é uma camada totalmente ligada, semelhante a uma rede neuronal multicamada *perceptron*, que realiza o mapeamento entre as características latentes e as classes de saída. O output desta camada é regulado pela função *softmax*:

$$v_{ij} = \frac{\exp(v_{(i-1)j})}{\sum_{j=1}^C \exp(v_{(i-1)j})}$$

onde, C é o número de classes de saída. Esta função providencia a probabilidade da classificação dos resultados.

Para reverter as predições matriciais resultantes da aplicação da CNN, os resultados sofrem um processo em dois passos, na primeira fase os resultados são categorizados pela mesma categoria identificada ao nível matricial e, em seguida, para os resultados que se sobrepõem, é utilizado um método de votação para inferir a categoria final desta amostra. Uma consequência da dependência temporal dos sinais sensoriais é que as categorias inferidas muitas vezes têm uma tendência suavizada, que pode ser melhorada através da aplicação de um filtro passa-baixo para remover ruído e manter os extremos que permitem identificar as mudanças de atividade, o que pode permitir uma inferência muito mais eficaz.

2.6. Considerações finais do capítulo

Neste capítulo realizou-se uma revisão bibliográfica sobre as temáticas base desta tese, de forma a definir as bases que nos permitem orientar a nossa investigação futura. Neste capítulo, apresentou-se extensamente a dificuldade da definição de contexto, as diversas propostas de vários autores e a evolução em busca de uma definição geral que permitisse criar a verdadeira definição de contexto.

Nessa secção foi apresentada também a nossa definição de contexto e a razão pela qual, na nossa opinião, a definição mais utilizada não se enquadra corretamente, sendo por isso, necessário estender a definição de acordo com as nossas reais necessidades. Foram apresentadas, também, já algumas definições que nos irão guiar, como sejam, a definição de categorização contextual, o atributo contextual e o modelo contextual, permitindo assim clarificar as nossas opções futuras.

Posteriormente, apresentou-se a temática da sensibilidade ao contexto e de que forma se considera que uma aplicação é sensível ao contexto. Conforme a definição anterior, foram também apresentadas as diferentes visões de vários autores e de que forma a definição que iremos utilizar foi evoluindo. Por fim, relativamente à temática de contexto, foi realizada uma revisão sobre o ciclo de vida do contexto. Esta revisão tem como objetivo adicionar um nível de compreensão base sobre o seu ciclo de vida e as principais propostas para cada tipo de categoria.

No Capítulo 3, apresenta-se uma revisão de estado da arte ao nível de *middleware* contextual que permita utilizar o contexto em aplicações móveis e de que forma estas influenciaram o desenvolvimento do *middleware* ADAPT.

3. ESTADO DA ARTE DE MIDDLEWARE CONTEXTUAL

Este capítulo começa por introduzir o Context Toolkit (Dey, Abowd, & Salber, 2001), uma referencia conceptual, para o desenvolvimento de sistemas de middleware sensíveis ao contexto para dispositivos móveis. Este modelo é usado para a análise de diversos sistemas, académicos (e comerciais), que são apresentados e analisados neste capítulo.

3.1. Enquadramento

Weiser (1999) propôs, na sua visão para o computador do século XXI, que este computador seria capaz de compreender o contexto do utilizador e disponibilizar um conjunto de serviços e aplicações relevantes introduzindo, assim, a temática dos sistemas ubíquos e a sua relação com o contexto. Os sistemas ubíquos teriam desse modo, como requisito, a sensibilidade ao contexto, sendo que o principal desafio, na visão do autor, o desenvolvimento dessa sensibilidade ao contexto.

Com vista a simplificar o esforço no desenvolvimento de sistemas complexos, que necessitem de suportar um conjunto diversificado de equipamentos e tecnologias, é comum o desenvolvimento de sistemas intermédios, ou *middleware*, que se apresentam como uma camada de abstração, para permitir o desenvolvimento mais eficaz e promover a reutilização de técnicas e componentes. Também, no desenvolvimento de sistemas sensíveis ao contexto, existe esta necessidade, não só pela diversidade de sensores e configurações de hardware disponíveis, mas também pelas restrições computacionais existentes, o que leva a que seja necessário reutilizar a de informação de contexto em várias aplicações.

A proposta conceptual Context Toolkit, proposta por (Dey et al., 2001) é considerada por muitos como a base para o desenvolvimento de aplicações sensíveis ao contexto. Nesta proposta, a aquisição e a representação da informação contextual é separada da sua utilização por parte de uma aplicação sensível ao contexto. O Context Toolkit propõe as seguintes orientações para o desenvolvimento de um *middleware* que lide com o contexto:

- *Separação de responsabilidades* – Cada sensor deverá recolher informação, independentemente de onde é usado ou de como é usado bem como o estado de funcionamento de outros sensores. A interrupção do funcionamento do sensor não deverá perturbar o funcionamento de outros.
- *Interpretação contextual* – A utilização de múltiplas camadas de contexto deve ser transparente e suportada pela arquitetura.

- *Identificação de recursos* – para uma aplicação comunicar com um sensor, ou fonte de contexto, este deve disponibilizar que tipo de informação pode providenciar, onde se encontra, e como comunicar com ele.
- *Comunicação transparente e distribuída* – a comunicação entre os sensores e as aplicações deve ser transparente e com suporte para múltiplas utilizações, por diversos produtores e consumidores de contexto.
- *Disponibilidade da aquisição de contexto constante* – Os sensores contextuais devem operar independentemente das aplicações que os consumam, e o estado destas não deve perturbar a execução de outras (i.e., se a aplicação terminar abruptamente, o sensor contextual não deve apresentar diferenças de funcionamento e vice-versa).
- *Armazenamento de contexto* – O histórico contextual pode (e deve) ser utilizado para estabelecer tendências e prever futuros valores contextuais. Assim, deve ser armazenado localmente, de forma segura, e opcionalmente após anonimização dos dados, enviado para um servidor remoto para inferência de contexto computacionalmente mais complexo.

Outro detalhe fundamental num *middleware* contextual é a reutilização de contexto, e com este efeito, o *Context Toolkit*, propõe ainda:

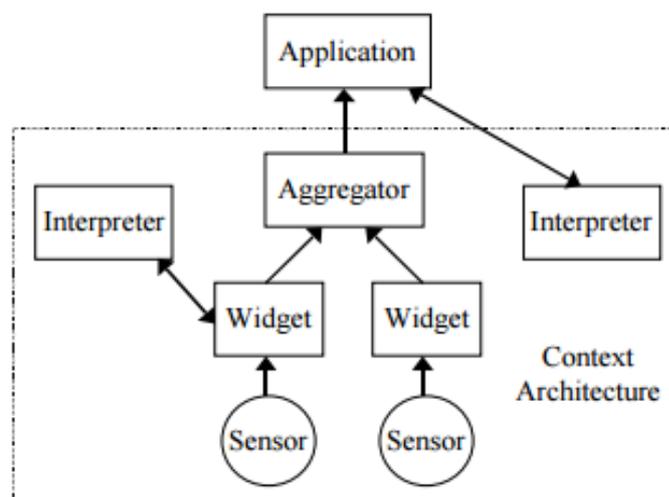


FIGURA 3-1 – MODELO DE UTILIZAÇÃO DE CONTEXTO (DEY ET AL., 2001)

- *Context Widgets* – Blocos de informação contextual, que permitem abstrair da complexidade de interação com o sensor e disponibilizam a informação num formato que corresponde às expectativas da aplicação.
- *Interpreters* – são sensores virtuais que utilizam técnicas de inferência a partir da informação de um ou mais sensores e produzem uma nova informação contextual.
- *Aggregators* – são grupos de sensores que contêm informação que está associada entre si.
- *Services* – São componentes que executam ações pelas aplicações, de forma síncrona ou assíncrona.
- *Discoverers* – São responsáveis por manter o registo das capacidades que estão disponíveis.

A informação contextual não é só baseada em sensores e no processamento desses dados, mas também baseada no utilizador, que é o principal fornecedor de contexto e também o seu utilizador. A secção seguinte destaca as principais propostas no desenvolvimento de uma *middleware* contextual para dispositivos móveis, que consideram sensores físicos, inferidos, mas também baseados no utilizador. São apresentadas por ordem cronológica.

A seleção destas propostas foi realizada através da seguinte metodologia:

3. Construção de uma base de dados de propostas científicas utilizando os seguintes parâmetros “*Context-Aware**”, “*Ubiquitous computing*”, “*middleware OR framework*”, “*mobile*”, utilizando as bases de dados Web of Knowledge (<http://webofknowledge.com>), Science Direct (<http://www.sciencedirect.com/>), IEEE Explore (<http://ieeexplore.ieee.org/>), e Google Scholar (<https://scholar.google.pt/>). Os resultados foram importados para o programa *Publish or Perish* (Harzing, 2007), uma aplicação gratuita que permite obter métricas sobre os artigos de forma simples, nomeadamente, ao nível de citações

total, ajustado ao ano, e detecção automática de duplicados, entre outras. Após a limpeza de duplicados, compilou-se uma lista de 1326 artigos, dos quais foram removidos 372 por não possuirmos acesso ao documento.

4. Com base na lista de 954 artigos, procedemos à exclusão de propostas não diretamente relevantes para os requisitos: “ser uma proposta de middleware, ou que não permitam reutilizar e partilhar o contexto adquirido”, “seja relativo a aplicações móveis ou aplicáveis a dispositivos móveis”. Esta exclusão foi realizada por análise dos títulos e resumos dos artigos e, em caso de dúvida, por análise do artigo, tendo sido removidos os artigos de tipo: *Survey (102)*, *Revisões Teóricas ou position papers (135)*, *Não é uma proposta de middleware (345)*, *não relacionadas com aplicações móveis (156)*. Após esta análise, obteve-se uma lista de 216 artigos.
5. Seleção de um conjunto reduzido de propostas, a partir dos 216 artigos, com base nas seguintes regras:
 - *Ordenação por número de citações na literatura ajustada ao ano* (calculada pelo programa Publish or Perish de Harzing (2007)) – permitiu obter um conjunto de propostas que já foram analisadas e validadas por outros autores, e remover o enviesamento provocado pelo tempo de publicação;
 - *Relevância para a temática* – permitiu obter um conjunto de propostas que correspondem às necessidades, nomeadamente que sejam inseridas dentro da temática dos dispositivos móveis, ou, se possam aplicar aos mesmos – *excluídos 159 artigos por não relevância, 9 por não possuir acesso ao artigo.*
 - *Seleção por implementação das orientações propostas por* (Dey et al., 2001) – permitiu a inclusão apenas de artigos que seguem as diretivas

propostas por Dey para o desenvolvimento de middleware contextual – *excluídos 37 artigos*.

- *Inclusão de soluções comerciais* – permitiu incluir soluções que inicialmente começaram por ser propostas académicas e que evoluíram para soluções comerciais – *incluídas 3 propostas*;

Com base nesta metodologia de seleção, obteve-se uma amostra de 14 propostas académicas e comerciais, que são, em seguida, apresentadas e analisadas à luz das orientações propostas por (Dey et al., 2001).

3.2. CORTEX

O sistema CORTEX,(Biegel & Cahill, 2004), permite aos investigadores fundir a informação de vários sensores móveis, representando o contexto aplicacional e o raciocínio sobre o contexto. Incorpora o STEAM (Meier & Cahill, 2003), um protocolo de comunicação baseado em eventos para comunicações sem fios, de forma a suportar uma comunicação não dependente e restritiva entre sensores, atuadores e componentes aplicacionais.

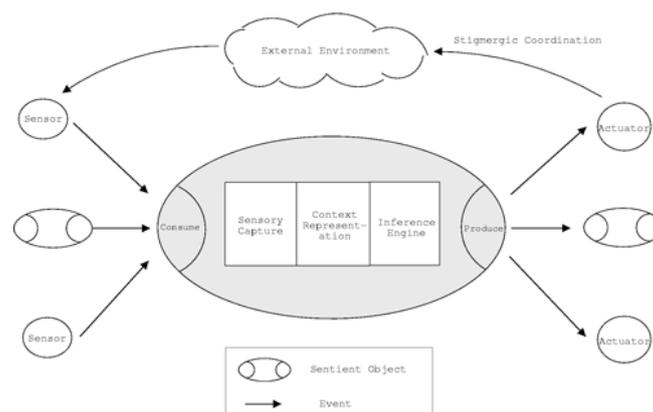


FIGURA 3-2 – MODELO CONCEPTUAL DO SISTEMA CORTEX (BIEGEL & CAHILL, 2004)

Este sistema introduz o conceito de objeto senciente no desenvolvimento de aplicações sensíveis ao contexto. Um objeto senciente é uma abstração lógica para os sensores e atuadores com as seguintes propriedades:

- *Senciente* – a capacidade de interpretar o estado do ambiente via sensores;
- *Autónomo* – a capacidade de operar independentemente do controlo humano e de forma descentralizada;
- *Proactivo* – a capacidade de agir de forma antecipatória de acordo com objetivos ou obstáculos.

Pela combinação de objetos sencientes e de um protocolo de comunicação sem fios baseada em eventos, o CORTEX, requer que os implementadores definam as entradas e saídas, contextos, os serviços de fusão e as regras através da utilização do modelo de inferência CLIPS (CLIPS, 1985). A inferência irá posteriormente seguir um modelo de execução evento-condição-ação (ECA) (de Ipiña & Katsiri, 2001)

3.3. Context Studio

O sistema Context Studio (Korpipää, Häkkinen, Kela, Ronkainen, & Känslä, 2004), é um *middleware* que tem em conta a mediação e responsabilidade do utilizador na inferência do contexto, pela dificuldade de automatizar completamente as ações baseando-se, apenas, no contexto inferido (Bellotti & Edwards, 2001)

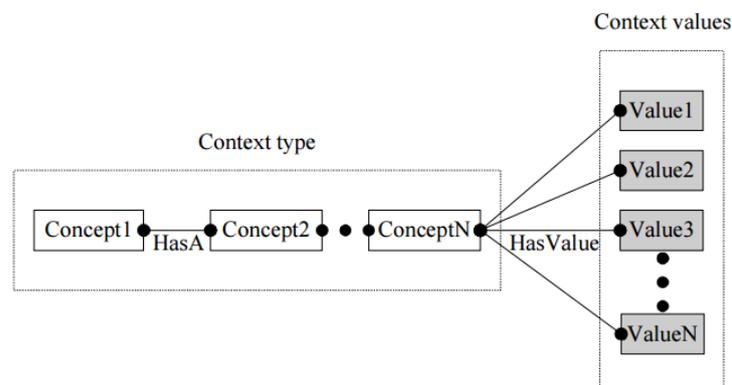


FIGURA 3-3 – MODELO CONCEPTUAL PARA CONSTRUÇÃO DE VOCABULÁRIOS DO CONTEXT STUDIO (KORPIPÄÄ ET AL., 2004)

Assim, o Context Studio considera a mediação das ações dependentes de contexto como manual, semiautomáticas, e totalmente automatizadas. As ações manuais são ações controladas pelo utilizador, baseadas no contexto atual. Semiautomáticas, são as ações que são baseadas num contexto inferido pelo dispositivo móvel e, mais tarde, ajustadas e confirmadas pelo utilizador. Totalmente automáticas são ações que são pré-programadas de acordo com o contexto detetado pelo dispositivo.

O Context Studio, utiliza uma abordagem baseada em dividir e conquistar, isto é, divide o problema em múltiplos subproblemas, que combinados resolvem o problema, para a criação de regras contextuais, ações, eventos e o respetivo resultado. Os utilizadores podem combinar as regras contextuais já existentes para gerar ações e eventos por parte do dispositivo e, assim, adicionar ao dispositivo sensibilidade ao contexto.

3.4. ContextPhone

O sistema ContextPhone (Raento, Oulasvirta, Petit, & Toivonen, 2005), é um *middleware* baseado em *widgets* para dispositivos móveis. Um *widgets* é uma pequena aplicação com apenas um propósito simples e totalmente funcional. O ContextPhone é construído baseando-se em quatro principais componentes: sensores, comunicação, *widgets* e serviços.

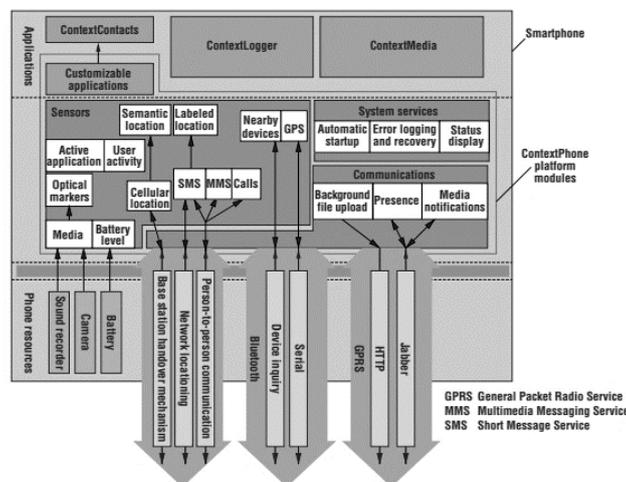


FIGURA 3-4 – ARQUITETURA DO SISTEMA CONTEXTPHONE (RAENTO ET AL., 2005)

Os sensores disponíveis interpretam a localização, a interação do utilizador, os comportamentos de comunicação e o ambiente físico. O conceito fundamental do ContextPhone assenta na ideia do contexto como um recurso para os utilizadores, ou seja, a compreensibilidade do contexto. Pela utilização de comunicações (ex. chamadas, mensagens), e serviços do sistema (ex. *bluetooth*, *wifi*), este sistema permite partilhar a localização e a proximidade a outros contextos via HTTP para um servidor remoto, e pela utilização de *widgets*, os utilizadores possuíam o controlo sobre a recolha dos dados.

3.5. AWARENESS

O sistema AWARENESS (van Sinderen, van Halteren, Wegdam, Meeuwissen, & Eertink, 2006), é um *middleware* que prioriza as questões de privacidade dos utilizadores. O *middleware* aplica o conceito de Qualidade de Contexto (*QoC*) para expressar a qualidade das características da informação contextual (Buchholz & Schiffers, 2003).

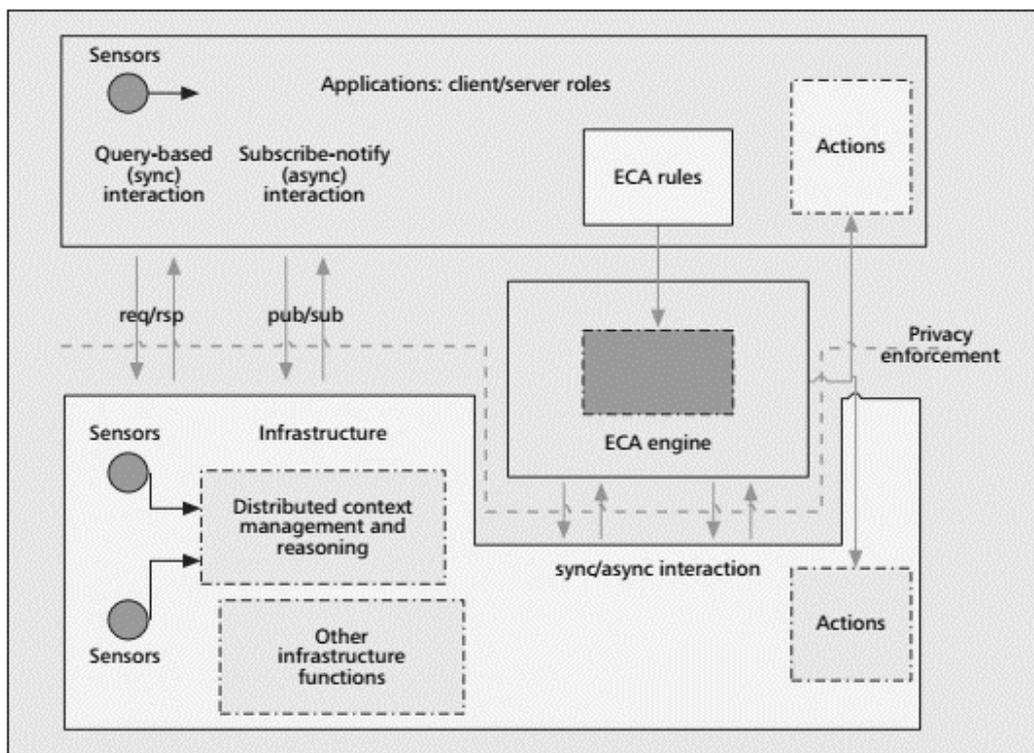


FIGURA 3-5 – ARQUITETURA DO SISTEMA AWARENESS (VAN SINDEREN ET AL., 2006)

A preocupação da privacidade irá ter influência no QoC, através do seu aumento ou diminuição, dependendo em quanto o contexto é partilhado, num determinado tempo. (exemplo: utilizar a localização por WiFi invés de GPS para redução da precisão do contexto da localização). Além disso, o contexto só será partilhado com dispositivos previamente autorizados, sendo o utilizador o controlador dos aspetos de privacidade. O sistema AWARENESS tem como alvo o desenvolvimento de aplicações baseadas na saúde, para pacientes e investigadores médicos.

3.6. MyExperience

O sistema MyExperience, (Froehlich, Chen, Consolvo, Harrison, & Landay, 2007), é um *middleware* que captura a informação de sensores e do utilizador, de forma a tentar compreender as motivações, perceção e satisfação do utilizador pela tecnologia. A recolha de dados do utilizador tinha por base a resposta a questionários, sendo a avaliação da experiência de utilizador efetuada com base em eventos de sensores e de regras pré-estabelecidas, por parte dos utilizadores.



FIGURA 3-6 – ARQUITETURA CONCEPTUAL DO SISTEMA MYEXPERIENCE (FROEHLICH ET AL., 2007)

Estes eventos podem ser: iniciar uma aplicação, apresentar uma notificação, reproduzir música, vibrar, piscar com o flash, realizar uma captura de ecrã, enviar SMS, ou apresentar um questionário. O myExperience utiliza uma abordagem de sincronização remota, isto é, de forma a permitir o acesso expedito à informação, assim que possui uma ligação de dados, envia automaticamente toda a informação que necessita.

3.7. CenceMe

O sistema CenceMe (Miluzzo et al., 2008), é um *middleware* que infere a presença de indivíduos através da utilização de sensores na aplicação móvel e partilha de informação através de uma rede social (ex. Facebook).

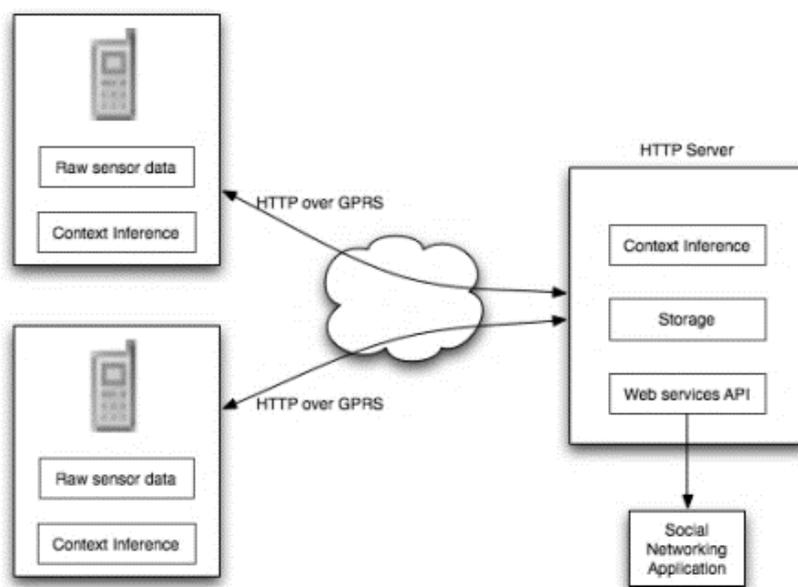


FIGURA 3-7 – ARQUITETURA CONCEPTUAL DO MODELO CENCEME (MILUZZO ET AL., 2008)

O CenceMe, introduz uma classificação repartida para partilhar o contexto social. Este contexto é detetado localmente no dispositivo e transferido para um *backend* para ser realizada uma correspondência com contextos sociais comuns, de forma a aumentar a consciência social.

Com a classificação repartida, parte pode ser realizada no equipamento móvel, com o suporte do *backend*, ou apenas por este. De forma a inferir os eventos sociais, é utilizado o

acelerómetro, Bluetooth, áudio e GPS. O GPS e o acelerómetro permitem detetar possíveis atividades do utilizador, ex., estar sentado, em pé, a andar ou a correr; o Bluetooth, permite a localização de outros dispositivos móveis; a análise de áudio permite detetar a presença de pessoas a falar.

Os dados capturados são enviados para o servidor, através de mensagens XML, para classificação social. Do lado do servidor, um classificador realiza o processamento das mensagens para inferir um de 4 estados sociais: a conversar, a dançar, numa festa, ou sozinho. O CenceMe, foca-se: na experiência social dos utilizadores através da partilha de informação de presença; em estimular a curiosidade, entre os utilizadores, enquanto se movem; e aprender com base nos seus padrões e movimento.

3.8. SociableSense

O sistema SociableSense (Rachuri et al., 2010; Musolesi et al., 2011), foca-se no contexto social psicológico. Este *middleware* permite inferir emoções individuais, atividades e interações verbais e de proximidade entre amigos. Este sistema pode detetar a identidade de quem está a falar, as emoções e a localização através da utilização dos sensores, microfone, Bluetooth e GPS. Um motor de inferência permite adaptar a taxa de amostragem, de acordo com o estado do utilizador (ex. parado, em movimento), e o que o rodeia (ex. pessoas à sua volta, conversas), de forma a minimizar o consumo de energia durante a captura e processamento, mas mantendo um nível de precisão de inferência elevado.

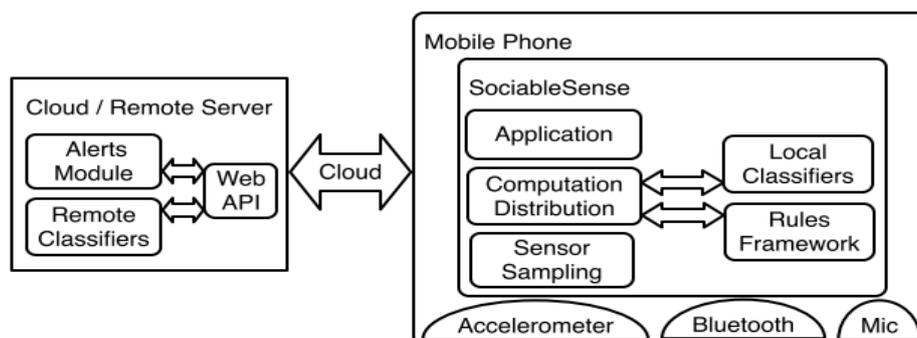


FIGURA 3-8 – ARQUITETURA CONCEPTUAL DO SISTEMA SOCIABLESENSE (RACHURI ET AL., 2010)

A identificação do utilizador e da emoção deste, é realizada no dispositivo móvel através de um classificador *Gaussian Mixture Model*, previamente treinado com os participantes do estudo, e implementado utilizando o *Hidden Markov Model Toolkit* (HTK, 2016). O SociableSense tem como alvo os investigadores com interesse nas interações sociais, permitindo descrever regras de sensorização e gerir os sensores de acordo com o contexto social dos utilizadores.

3.9. Empath

O sistema Empath (Emotional Monitoring for PATHology) (Dickerson, Gorlin, & Stankovic, 2011), é um *middleware* para supervisionar a saúde emocional para doença depressiva. Este sistema é composto por um conjunto de sensores sem fios, um ecrã de toque, e dispositivos móveis. Os diagnósticos dos pacientes e o planeamento terapêutico é suportado por relatórios gerados pela agregação de contexto como o sono, peso, atividades diárias, e a métrica de fala.

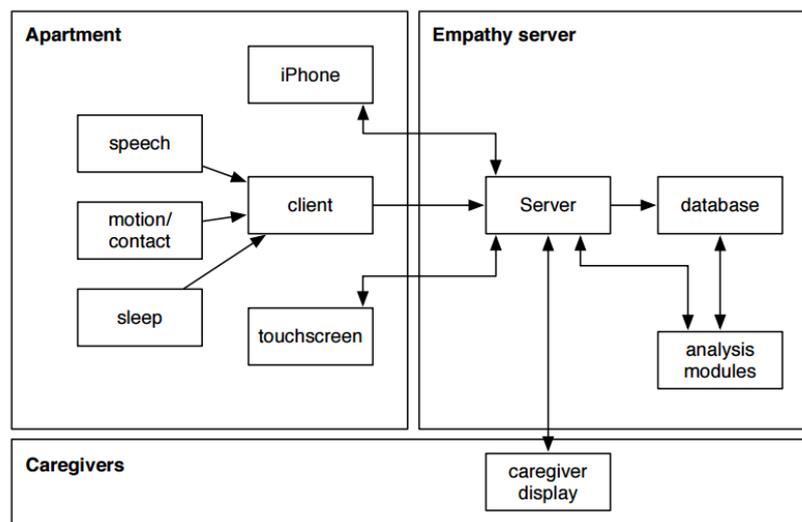


FIGURA 3-9 – ARQUITETURA CONCEPTUAL DO SISTEMA EMPATH (DICKERSON ET AL., 2011)

Os sensores são distribuídos por diferentes localizações, em casa dos pacientes e no seu dispositivo móvel, e os dados são recolhidos localmente. A sincronização agendada em intervalos diários, semanais, ou bissemanais, envia os dados para uma base de dados, num

servidor web remoto. A análise de comportamentos, é executada no servidor, e os resultados são disponibilizados nos ecrãs de toque, na casa dos utilizadores.

3.10. Funf

O sistema Funf (Friends and Family) (Aharony, Pan, Ip, Khayal, & Pentland, 2011), foca-se na inferência social e comportamental. Este sistema utiliza os sensores de hardware e software disponíveis nos dispositivos móveis como: GPS, acelerómetro, Bluetooth, sistemas de comunicação (ex. chamadas, mensagens), aplicações instaladas, aplicações em execução e informação multimédia, entre outros. A leitura dos dados dos sensores é agendada em períodos de tempo, configuráveis, com uma duração (período de tempo mínimo que o sensor será executado), período (tempo entre as leituras de dados do sensor), início (data a partir da qual é autorizada a leitura) e fim (data a partir da qual os dados não são autorizados interpretar).



FIGURA 3-10 – MODELO CONCEPTUAL DO SISTEMA FUNF (AHARONY ET AL., 2011)

Os dados são armazenados e cifrados, localmente no dispositivo móvel, sob a forma de várias bases de dados *SQLite* fragmentadas no tempo. Os dados são acessíveis via decifragem e fusão posterior numa aplicação de desktop, que os converte para um formato mais fácil de processar. O público-alvo deste sistema são investigadores interessados em recolher dados sociais e comportamentais. Este sistema também pode ser utilizado por utilizadores que pretendam manter um registo detalhado do seu dia-a-dia.

3.11. SystemSens

O sistema SystemSens (Falaki, Mahajan, & Estrin, 2011), é um *middleware* que captura a utilização do contexto de uso de dispositivos móveis, onde o contexto é definido como sendo as interações dos utilizadores com a aplicação.

Estas interações podem incluir o consumo energético, chamadas, uso de processador, localização de célula de rede, estado de dados móveis e informação de chamadas. Os dados capturados são enviados em formato JSON via HTTP para um servidor web. De forma a aumentar as capacidades do *middleware*, os novos componentes podem ser adicionados via implementação de uma interface AIDL (*Android Interface Definition Language*).

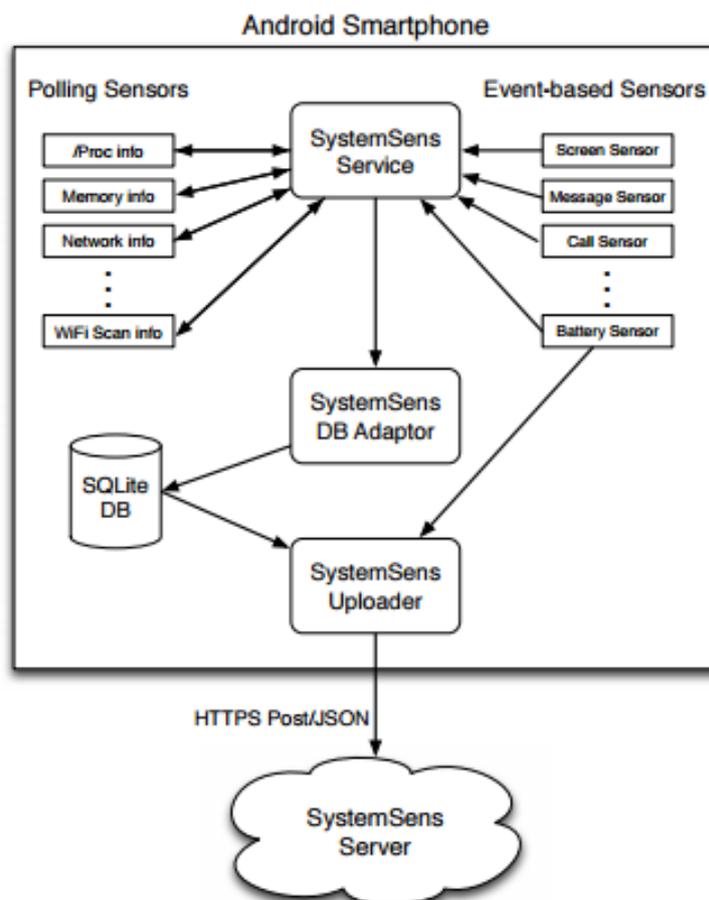


FIGURA 3-11 – ARQUITETURA DO SISTEMA SYSTEMSENS (FALAKI ET AL., 2011)

3.12. OpenDataKit

O sistema de sensores OpenDataKit (ODK) (Brunette et al., 2012; Open Data Kit, 2016), é um *middleware* que ambiciona simplificar a interface entre os sensores externos e os dispositivos móveis. O ODK permite a abstração do desenvolvimento de aplicação e sensores, das aplicações do utilizador e dos drivers que consomem essa informação, através da gestão de um registo de capacidades, canais de comunicação e buffer de dados.

É uma arquitetura baseada em componentes, que permitem aos desenvolvedores aplicativos focarem-se no desenvolvimento da sua aplicação, requerendo o mínimo de código possível para a integração de sensores. Assim, ao evitar que estes tenham que escrever código específico para cada tipo de sensor, permite criar um ecossistema de sensores reutilizáveis e multi-hardware.

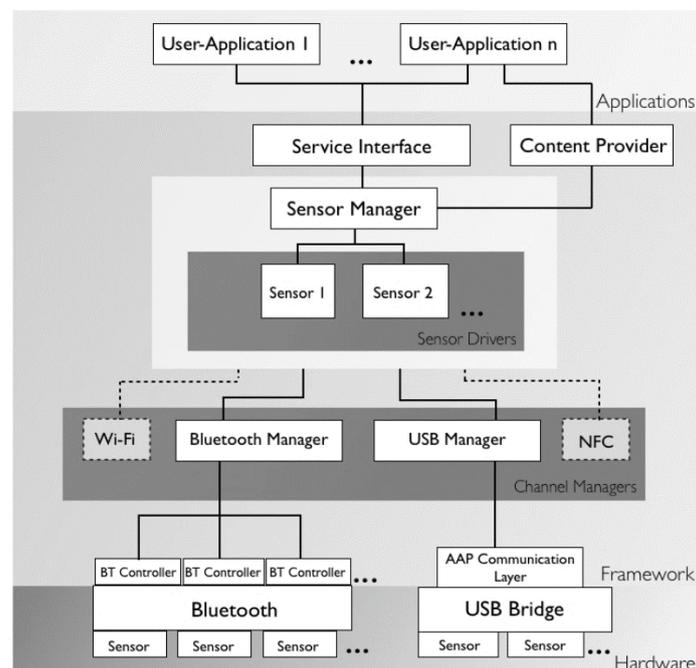


FIGURA 3-12 – ARQUITETURA CONCEPTUAL DO SISTEMA OPENDATAKIT (BRUNETTE ET AL., 2012)

Com esta abordagem é possível disponibilizar um elevado nível de isolamento, entre as aplicações e o código específico sensorial, permitindo que estas funcionem mesmo em situações em que os sensores não funcionam corretamente, ou mesmo, não se encontram disponíveis. A

integração de novos sensores na aplicação é realizada através do download de novos sensores, a partir de um mercado aplicacional sem ser necessário quaisquer alterações do sistema operativo.

3.13. Ohmage

O sistema ohmage (Tangmunarunkit et al., 2015), é um *middleware* que regista, analisa e visualiza dados obtidos por pedidos de informação ao utilizador, e também por registo contínuo da informação dos sensores, com vista à sua utilização em aplicações relacionadas com a saúde.



FIGURA 3-13 – ARQUITETURA CONCEPTUAL E MODULAR DO SISTEMA OHMAGE (TANGMUNARUNKIT ET AL., 2015)

O Ohmage é uma plataforma móvel e web, desenhada para criar e gerir a amostragem de experiências, baseando-se na recolha de dados de questionários, direcionados para a saúde, com dispositivos móveis. O *middleware*, suporta: eventos baseados em tempo e localização para a recolha de dados via questionários; reconhecimento de atividades baseadas em fusão de sensores, como, o acelerómetro, GPS, Wi-Fi, e informação de rede móvel (BTS CID), registo de horas de sono e de exercício; registo de interação social baseado na deteção e processamento de áudio, mensagens motivacionais para motivar o utilizador. Os dados são enviados para um servidor seguro remoto, onde podem ser consultados.

O sistema Ohmage foi avaliado em sete estudos independentes, com diferentes tipos de população, incluindo: mães pela primeira vez, adolescentes, mulheres emigrantes, sobreviventes de cancro da mama, doentes seropositivos, pessoas com défice de atenção e hiperatividade e doença de stress pós-traumático.

3.14. AWARE

O sistema AWARE (Ferreira, Kostakos, & Dey, 2015), é um *middleware* open-source que tem como principal objetivo a criação de uma plataforma extensível e reutilizável para a sensorização, inferência, e geração de contexto em dispositivos móveis. Este sistema apresenta uma abstração ao nível da implementação específica dos sensores, o que permite a quem utilize este *middleware*, não tenha que se preocupar com implementações específicas para os diferentes tipos de hardware disponíveis. Os dados recolhidos são armazenados localmente, mas podem, para a utilização em casos de estudo mais alargados, ser enviados para um servidor remoto, para posterior processamento e análise.

O sistema AWARE possui uma interface destinada ao utilizador do dispositivo móvel, que lhe permite controlar o nível de partilha dos dados, ativar, desativar ou instalar novos sensores, mantendo assim a capacidade de controlo da privacidade do utilizador.

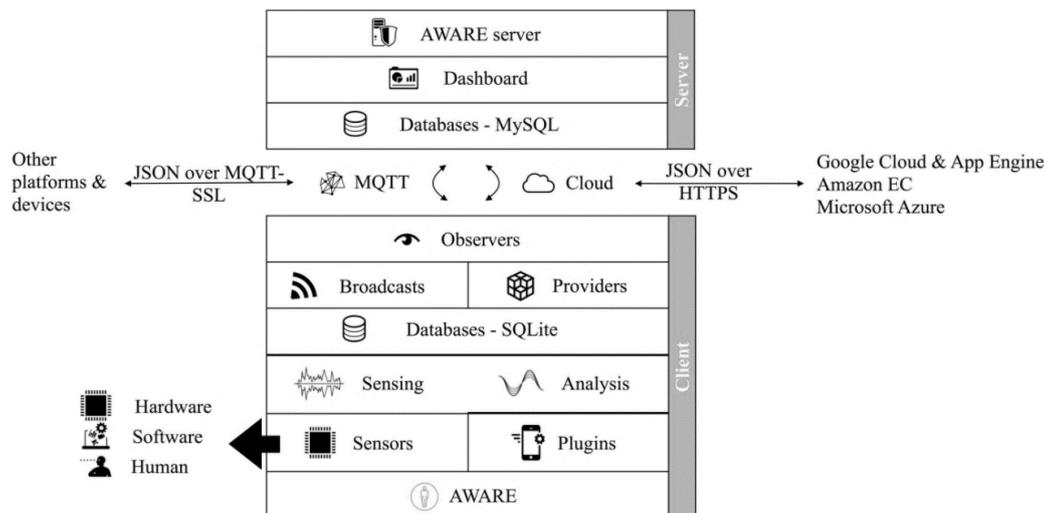


FIGURA 3-14 – ARQUITETURA CONCEPTUAL DO SISTEMA AWARE (FERREIRA ET AL., 2015)

Os dados sensoriais são partilhados com as aplicações, que os pretendam usar, sob 3 formas:

- *Broadcasts* – Para atualizações rápidas da informação contextual em tempo real, cada evento possui uma breve descrição do contexto do utilizador e os dados capturados correspondentes a esse contexto. Devido à natureza de funcionamento dos *broadcasts*, é possível uma aplicação receber em simultâneo, eventos de vários sensores.
- *Provider* – Para armazenamento de dados dos sensores e informação de *plugins* com histórico. Os dados são armazenados localmente ou, se necessário, remotamente. É possível efetuar um pedido diretamente a um *provider*, ou através de um *observer*.
- *Observers* – Monitoriza as alterações dos sensores e do contexto em tempo real, partilhando os mesmos com outros dispositivos, via um sistema *de message queue telemetry transport (MQTT)*. Os *observers* disponibilizam acesso ao contexto de forma ativa (via *push*) ou baseado em eventos, são a abordagem mais eficiente ao nível energético.

3.15. Ginger.io

O sistema Ginger.io (Ginger Inc, 2016), é um *middleware* para a análise de sintomatologia psicopatológica, e respetivo acompanhamento por profissionais qualificados. O sistema Ginger.io converte a informação sensorial do dispositivo móvel em conhecimento, que será posteriormente utilizado para gerar um relatório pessoal do utilizador. O Ginger.io divide-se em duas partes, uma aplicação de gestão web para os investigadores responsáveis, e uma aplicação móvel para os pacientes. A aplicação móvel recolhe a informação de forma passiva, incluindo o movimento, padrões de chamadas, envio e receção de mensagens.



FIGURA 3-15 – MODELO CONCEPTUAL DO PROJETO GINGER.IO (GINGER INC, 2016),

Dependendo do pretendido pelo investigador responsável, a aplicação móvel pode requerer, diária ou semanalmente, informação do utilizador sob a forma de um questionário de pequena dimensão.

3.16. Considerações finais do capítulo

O desenvolvimento de sistemas sensíveis ao contexto é uma ação significativamente complexa e de longa duração devido, não só à falta de suporte para o mesmo mas, principalmente, devido à diversidade de configurações, sensores, capacidades e precisão dos sensores disponíveis nos equipamentos móveis atuais. Esta revisão de estado da arte demonstra o quão fragmentada está a investigação em sensibilidade ao contexto, onde cada investigador se foca no seu domínio e nos requisitos deste.

Nesta tese, considerou-se a sensibilidade ao contexto pela forma como os sistemas constroem e adquirem o contexto. Isto permitiu separar os sistemas em duas categorias: *centralizado*, onde a gestão contextual é realizada apenas no dispositivo; *distribuído*, onde é permitido partilhar com outros dispositivos, ou delegar o processamento para outros sistemas.

No seguimento da proposta conceptual de Dey et al. (2001) o contexto pode possuir 3 propriedades:

- *Partilhado* – importante para a investigação multidisciplinar e colaboração, ao ambicionar resolver os problemas de reutilização, delegação e acessibilidade;
- *Dinâmico* – importante para dar resposta aos desafios de volatilidade, como a adaptação em tempo real, e a manipulação, como a regulação de frequência;
- *Dimensionável* – importante para os desafios de heterogeneidade, transparência, redundância e portabilidade do contexto.

Na Tabela 3-1, são apresentadas as soluções anteriormente analisadas e apresentadas, acrescentando-se ainda a seguinte métrica:

- *Público alvo* – são categorizadas em 3 categorias, investigação científica, desenvolvimento aplicacional, utilizador final.
- *Sensores* – origem da informação contextual, hardware, software ou utilizador.
- *Propriedades contextuais* – *partilhado* se o contexto pode ser reutilizado; *dinâmico* se pode ser modificado e se se adapta ao contexto atual; *dimensionável* se suporta o incremento de novas fontes sensoriais.
- *Funcionalidades* – *controlo*, se o utilizado tem algum tipo de controlo sobre a recolha ou sobre os sensores; *visualização*, se é possível visualizar os dados recolhidos; *extensibilidade*, se é possível incorporar novos sensores e fontes contextuais; *partilha*, se permite partilha dos dados com outros dispositivos; *administração remota*, se tem a capacidade de ser controlado remotamente por um servidor; *delegação*, se este necessita de um servidor externo para o processamento e inferência do contexto.
- *Tipo de Gestão* – como a gestão do contexto é realizada, se localmente (centralizada) ou distribuída remotamente (distribuída)

TABELA 3-1 – RESUMO DO MIDDLEWARE EXISTENTE SENSÍVEL AO CONTEXTO

MIDDLEWARE	PUBLICO ALVO		SENSORES			PROPRIEDADES			FUNCIONALIDADES					TIPO GESTÃO	
	INVESTIGAÇÃO APLICACIONAL	UTILIZADOR	HARDWARE	SOFTWARE	UTILIZADOR	PARTILHADA	DINÂMICA	DIMENSIONÁVEL	CONTROLO	VISUALIZAÇÃO	EXTENSIBILIDADE	PARTILHA	ADMINISTRAÇÃO		DELEGAÇÃO
CORTEX	X		X	X		X		X		X	X				CENTRALIZADA
CONTEXT STUDIO		X	X	X	X	X	X		X		X				CENTRALIZADA
CONTEXT PHONE	X	X	X	X	X	X			X	X		X			DISTRIBUÍDA
AWARENESS	X		X	X		X	X		X	X		X	X		DISTRIBUÍDA
MYEXPERIENCE	X		X	X	X				X				X		DISTRIBUÍDA
CENCEME		X	X	X			X			X		X		X	DISTRIBUÍDA
SOCIABLESENSE	X		X	X			X		X	X					CENTRALIZADA
EMPATH	X	X	X	X				X	X	X			X	X	DISTRIBUÍDA
FUNF	X	X	X	X	X	X		X	X	X	X				CENTRALIZADA
SYSTEMSENS	X		X	X				X		X	X	X	X	X	DISTRIBUÍDA
OPENDATAKIT		X	X			X	X	X				X			DISTRIBUÍDA
OHMAGE	X		X	X	X	X			X	X			X		CENTRALIZADA
AWARE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	MÚLTIPLA
GINGER.IO	X	X	X	X		X			X	X			X		DISTRIBUÍDA

Neste capítulo pretendeu-se apresentar as soluções de sensibilidade contextual ao nível de *middleware* com capacidade integrativa em aplicações. Com esse objetivo, efetuou-se uma recolha das principais propostas académicas e comerciais existentes com esse fim, apresentados aproximadamente na última década. Esta seleção foi realizada tendo em conta as orientações propostas por Dey et al. (2001), na sua proposta conceptual. Fora do âmbito desta tese encontram-se aplicações contextuais que não possuam a capacidade de reutilização de informação contextual, como aplicações que apenas utiliza contexto.

No capítulo seguinte, será apresentado o modelo de arquitetura ADAPT, o modelo de adaptação dinâmica e sensível ao contexto de interfaces móveis em ambientes ubíquos, onde se descreve, de forma introdutória, o modelo de adaptação contextual para a construção de aplicações móveis antecipatórias.

4. MODELO CONTEXTUAL ANTECIPATÓRIO

Este capítulo apresenta um modelo teórico que suporta o desenvolvimento de um middleware para adaptação contextual dinâmica em dispositivos móveis, utilizando a informação contextual de sensores físicos, virtuais, e informação relativa ao utilizador. Inicialmente serão expostas as motivações para a adoção deste modelo e as diretrizes que este deverá implementar, sendo discutidas as vantagens que este modelo apresenta.

4.1. Enquadramento

A evolução dos dispositivos móveis, associada ao constante aumento das capacidades computacionais, criou condições para o desenvolvimento de aplicações sensíveis ao contexto, que objetivam poder aumentar a eficácia, eficiência e conforto de utilização, em qualquer situação e em qualquer localização, independentemente do seu estado. A deteção do comportamento humano, depende da perceção do contexto, ambiente externo e do conhecimento de padrões anteriores e, por isso, diversa investigação tem sido dedicada à implementação de sistemas computacionais que permitam inferir atividades, a partir de modelos conceptuais de alto nível e de extração de contexto, a partir do uso de sensores físicos (e.g. Roggen et al., 2011; De Pessemier, Dooms, & Martens, 2013; Gordon et al., 2014; Lee & Cho, 2014).

Diversas propostas, como as referidas no Capítulo 3, apresentam a construção de um *middleware* para implementação de um modelo para o reconhecimento das atividades diárias do utilizador, através da informação recebida pelos diversos sensores disponíveis. A principal dificuldade prende-se com a inferência de um modelo contextual, a partir de dados obtidos diretamente dos sensores. Os sensores atualmente disponíveis, nos dispositivos móveis mais comuns (como por exemplo acelerómetro/giroscópio, GPS e microfone), permitem a deteção de um conjunto de atividades específicas (incluindo, respetivamente, a deteção de mudanças de orientação do dispositivo, a localização, e conversações de voz).

No entanto, pela introdução de inteligência, sensibilidade situacional, e reconhecimento de contexto nestes dispositivos, é possível aumentar as suas capacidades de forma a melhorar, proactivamente, o dia-a-dia do utilizador, utilizando os sensores existentes como, por exemplo, contagem de passos através do uso do acelerómetro do utilizador. Neste sentido, para que a arquitetura do sistema permita a associação do significado semântico aos dados, são fundamentais a utilização sensorial e a inferência contextual para a sua adaptação.

A definição comum de *middleware* para sistemas contextuais é similar às propostas convencionais, como a proposta pelo RFC 2768, que descreve o *middleware* como um serviço

que se encontra numa camada acima do transporte e abaixo da aplicacional no modelo OSI (Aiken et al., 2000). Esta definição transcreve-se para o nosso conceito, situando-se entre os sensores e a aplicação (Figura 4-1).

Deste modo, é possível lidar com a diversidade de sensores e configurações existentes e expandir as capacidades sensoriais, através da utilização de mecanismos e serviços mais especializados do que aqueles que o sistema operativo pode disponibilizar. Este nível de transparência permite à aplicação abstrair-se das implementações específicas das outras camadas e assim, poder receber contexto sem, no entanto, saber necessariamente qual a origem do mesmo. Desta forma, é possível retirar a complexidade necessária para a implementação desta técnica, permitindo que o esforço do desenvolvimento aplicacional seja concentrado na melhoria da própria aplicação e a sua respetiva lógica aplicacional, e que a responsabilidade pelos algoritmos, pela inferência e pela interação com os sensores seja tratada pelo *middleware*.

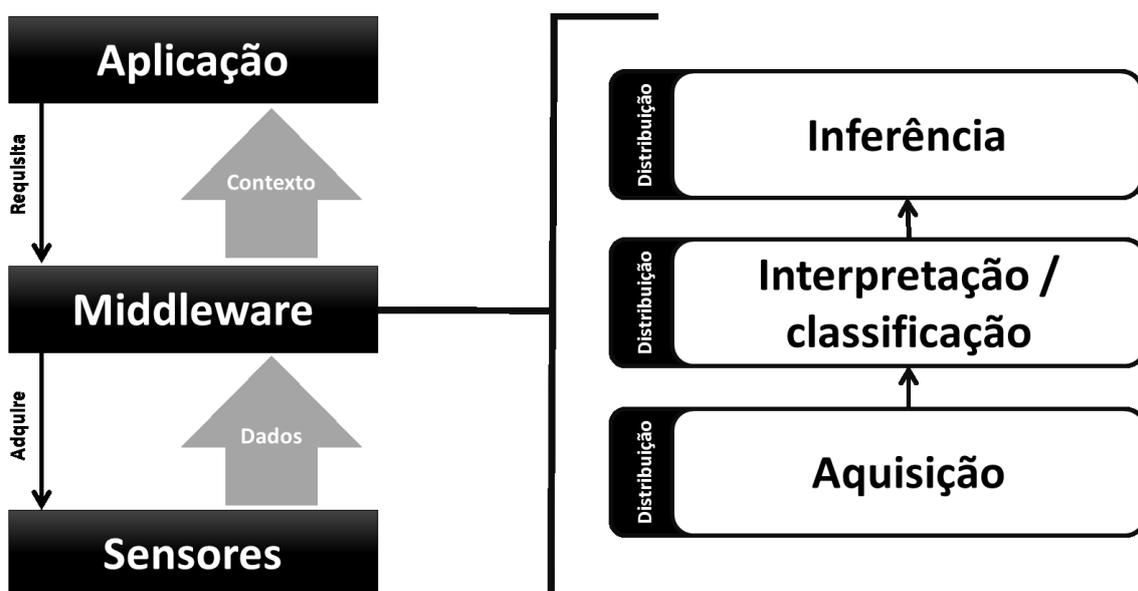


FIGURA 4-1 – ARQUITETURA CONCEPTUAL GENÉRICA DE UM *MIDDLEWARE* CONTEXTUAL

A Figura 4-1, apresenta a arquitetura conceptual ao nível das funcionalidades de um sistema de *middleware* contextual, representando o fluxo de informação e as fases pelas quais os dados passam para poderem permitir a construção da informação contextual. Para que a exploração do potencial contextual seja possível, é necessário obter a informação do contexto, que é recolhida por fontes de contexto, sejam eles físicos, (hardware), virtuais (software), ou

mesmo informação humana (utilizador). De forma a poder construir-se a aplicação, o mais eficientemente possível, é necessário que esta aquisição seja realizada da forma mais transparente possível, sendo que esta, no entanto, pode variar no formato, unidade, frequência e volume de dados.

Na fase seguinte deste processo é realizada a interpretação e ou classificação dos dados, responsáveis pela primeira fase de processamento, onde são obtidos alguns resultados com base na informação sensorial e com vista à obtenção de informação contextual direta, como o estado do utilizador, etc. Estas classificações podem ser obtidas por processos de amostragens temporais dos sensores e, com base nesses dados, classificando-os de acordo com a estratégia mais eficaz para o efeito, conforme apresentado no Capítulo 2.

A inferência, tipicamente a fase final do processo, consiste na aplicação de algoritmos e técnicas de inferências avançadas que permitem utilizar a informação histórica de diversos sensores para obter a informação contextual mais relevante e eficaz para o utilizador, não só através da informação existente, mas também pela predição de comportamentos e ações com base nessa informação podendo, assim, antecipar as necessidades do utilizador.

Em seguida, neste capítulo, iremos apresentar os principais desafios no desenvolvimento de soluções sensíveis ao contexto e as principais diretrizes orientadoras da apresentação e construção do modelo conceptual ADAPT.

4.2. Desafios e diretrizes orientadoras

O crescente aumento da capacidade e diversidade de sensores e a sua integração/utilização em dispositivos móveis, tem permitido aumentar a compreensão contextual do ambiente em redor ao utilizador. A Tabela 4-1 apresenta, de forma sumária, alguns dos principais desafios e preocupações relacionadas com o desenvolvimento de sistemas sensíveis recolhidos com base em todas as propostas analisadas para sensibilidade ao contexto apresentadas nos Capítulos 2 e 3.

TABELA 4-1 – PRINCIPAIS DESAFIOS NO DESENVOLVIMENTO DE *MIDDLEWARE* SENSÍVEL AO CONTEXTO (ADAPTADO DE FERREIRA, 2013).

ÁREA	DESAFIOS A ULTRAPASSAR
ENERGÉTICOS	OTIMIZAÇÃO DE TRANSMISSÕES RADIO PROCESSAMENTO ENERGETICAMENTE EFICIENTE AGENDAMENTO DE RECOLHA DE AMOSTRAS
SENSORIAL	SELEÇÃO DINÂMICA DE SENSORES AMOSTRAGEM ADAPTATIVA DIVISÃO OPORTUNISTA DE PROCESSOS SENSORIZAÇÃO TEMPORALMENTE VARIANTE
AQUISIÇÃO	CALIBRAÇÃO DE SENSORES DISTRORÇÃO E RUÍDO ALTERAÇÃO DE ORIENTAÇÃO DO DISPOSITIVO LOCALIZAÇÃO DO DISPOSITIVO (NO BOLSO, EM CIMA DA MESA, NA CARTEIRA)
INFERÊNCIA	PARADIGMA DE APRENDIZAGEM GENERALIZAÇÃO COMPLEXIDADE COMPUTACIONAL VERIFICAÇÃO DE REDUNDÂNCIA
MIDDLEWARE	CONTEXTO ASSÍNCRONO E HETEROGÊNEO INTEROPERABILIDADE ABSTRAÇÃO SENSORIAL NORMALIZAÇÃO TOLERÂNCIA A FALHAS TRANSPARÊNCIA DECENTRALIZAÇÃO ADAPTAÇÃO E CONFIGURAÇÃO AUTOMÁTICA
GERAIS	LIMITAÇÕES COMPUTACIONAIS, ENERGÉTICAS E ARMAZENAMENTO SEGURANÇA, PRIVACIDADE E CONFIANÇA ARQUITETURAS DE DISPOSITIVO COMPLEXAS E DIVERSAS FRAGMENTAÇÃO DE DISPOSITIVOS

Na Tabela 4-1, são apresentados alguns dos principais desafios que necessitam de resposta de forma a permitir a construção de um *middleware* eficaz e que cumpra o seu principal objetivo, a inferência contextual.

A fragmentação de dispositivos é um dos principais obstáculos pois, devido à multiplicidade e configurações de hardware, cada um tem sensores e capacidades diferentes,

levando a que o desenvolvimento de uma aplicação com estas capacidades, possa não apresentar os mesmos comportamentos nos diversos dispositivos.

Um estudo realizado em 2015 pela OpenSignal com 682 000 dispositivos Android, verificou um total de 24 093 modelos de dispositivos distintos, correspondendo a 1 294 fabricantes, que procederam à instalação da sua aplicação (OpenSignal, 2015). Na Figura 4-2, é possível visualizar a dimensão desse problema. Cada retângulo e respetiva área, correspondem a um modelo e à percentagem de cota de mercado. É, por este motivo importante que o sistema possua a capacidade de ser expandido facilmente, de forma a permitir o desenvolvimento de sensores adaptados às diversas características dos dispositivos.

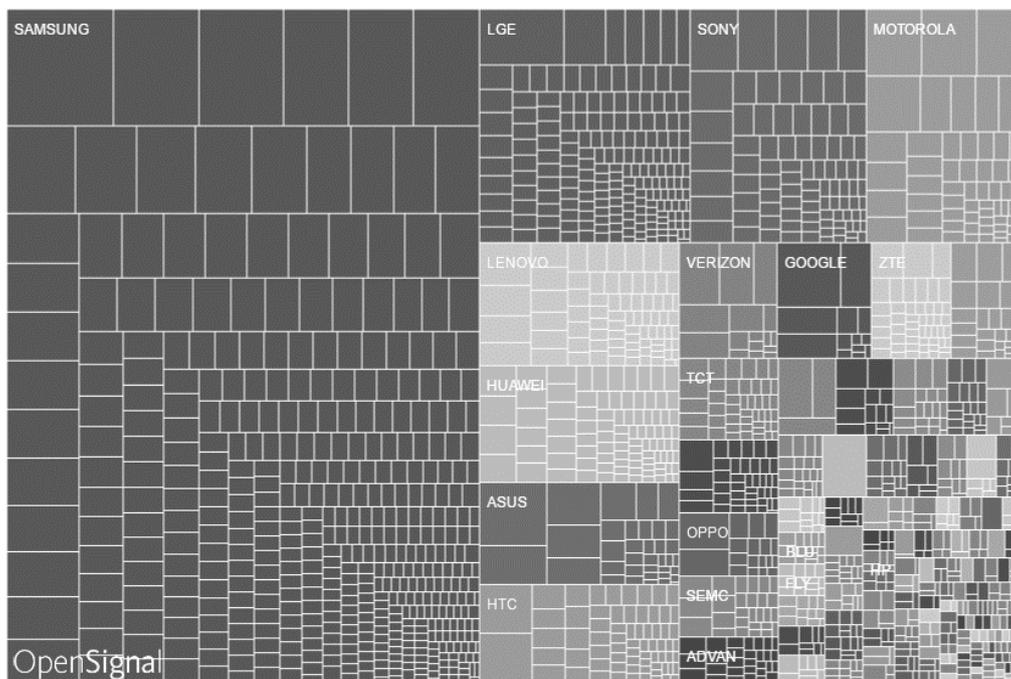


FIGURA 4-2 – FRAGMENTAÇÃO DE DISPOSITIVOS ANDROID REPARTIDO POR FABRICANTE (OPENSIGNAL, 2015)

O consumo energético associado à sensorização, é outra das preocupações que deve ser tida em linha de conta, pois, a utilização de sensores tem associado um consumo elevado dependendo do tempo e precisão associada. É, por isso, importante que o sistema tenha a capacidade de gerir, não só os sensores (necessários mediante o contexto), mas também a precisão e amostragem pretendida, para que o impacto na autonomia do dispositivo seja o menor possível, utilizando técnicas adaptativas e oportunistas de amostragem sensorial.

Uma arquitetura de *middleware* sensível ao contexto necessita de dar resposta a estes desafios, com um funcionamento transparente e descentralizado, evitando o contágio e propagação de erros de sensores isolados e, conseqüentemente, problemas nas aplicações que consomem os dados por eles gerados. Inspirado nos vários trabalhos apresentados no Capítulo 3 e seguindo como principal inspiração o modelo de referência *Context Toolkit* (Dey et al., 2001), num *middleware* contextual, o modelo deverá seguir as seguintes diretrizes (D):

- D-1. Permitir a adaptação e execução de ações automáticas utilizando as vantagens oferecidas pela inferência contextual em que o utilizador se encontra;
- D-2. Os serviços sensoriais e contextuais deverão resultar em serviços independentes, com *acoplamento fraco*, de modo que possam ser utilizados sozinhos ou por múltiplas aplicações;
- D-3. Oferecer ao utilizador uma forma de controlar o tipo e a sensibilidade contextual disponibilizada às restantes aplicações;
- D-4. Permitir o armazenamento seguro do histórico contextual para utilização futura no próprio dispositivo, ou com capacidade de exportação para um servidor para processamento posterior e partilha das conclusões;
- D-5. Fomentar a interoperabilidade e partilha da informação contextual entre aplicações que pretendam tornar-se sensíveis ao contexto;
- D-6. Permitir a expansibilidade das capacidades de inferência contextual no sistema de forma simples e parametrizável, conforme as necessidades.

4.3. Modelo de inferência contextual para um sistema antecipatório

Um *middleware* sensível ao contexto tem como objetivo fornecer às aplicações que pretendam utilizar os seus serviços, informação que lhes permita adaptar-se às necessidades do

utilizador disponibilizando, assim, uma melhor experiência de uso. A informação contextual do utilizador pode ser relativa ao seu meio físico, à interação, ao seu histórico, à sua agenda, ao seu objetivo, entre outros. Considera-se, assim, o contexto como todo o conjunto de informação e circunstâncias relevantes que possam ser utilizadas para representar a ação que o utilizador pretende realizar, incluindo também as restrições que esta pode sofrer em um determinado momento incluindo o histórico.

Para suportar a adaptação contextual sensível ao contexto para aplicações móveis em ambientes ubíquos é proposto o seguinte modelo de inferência contextual, representado pela Figura 4-3.

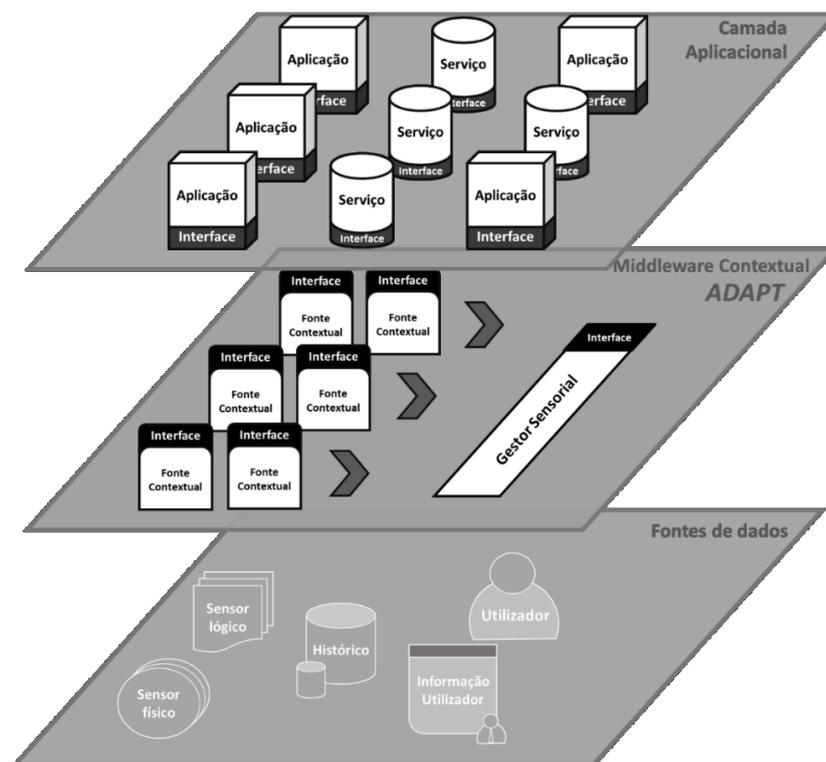


FIGURA 4-3 – ARQUITETURA CONCEPTUAL DO MODELO CONCEPTUAL ANTECIPATÓRIO

A arquitetura proposta é constituída pelos seguintes componentes:

- *Aplicações e Serviços* – Clientes que consomem a informação do modelo proposto para adaptar as suas interfaces e funcionalidades às necessidades do utilizadores, responsável por dar resposta à diretriz D-1

- *Fonte Contextual* – informação processada a partir dos dados dos sensores pela utilização de técnicas de inteligência artificial, a partir dos quais é possível tomar decisões ao nível de adaptação da aplicação sendo responsáveis pela implementação das diretrizes D-2, D-5 e D-6. Podem possuir os seguintes níveis de processamento:
 1. *Pré-processamento* – agregação e limpeza simples;
 2. *Interpretação* – processamento dos dados com base numa amostragem e posterior classificação dos resultados através de análise no tempo e na frequência;
 3. *Inferência* – previsão de comportamentos, ações e necessidades através da aplicação de técnicas de inteligência artificial.
- *Gestor Sensorial* – Sistema responsável pela manutenção do funcionamento do sistema, desde a manutenção do registo de sensores disponíveis e capacidades destes, à gestão de informação, privacidade e arquivo. É responsável pela implementação das diretrizes D-3, D-4, D-5 e D-6.
- *Fontes de dados* – origem dos dados em formato não processado, pode ter origem em diversos tipos de sensor: *físicos*, *lógicos* (sensores agregados, virtuais, inferidos), *informação histórica dos sensores*, *informação sobre o utilizador* (agenda, chamadas, perfil, favoritos) e, por fim, informação obtida diretamente do *utilizador*.

A arquitetura proposta ambiciona dar também resposta às orientações propostas por Dey et al. (2001), apresentando para cada uma das orientações, as seguintes soluções:

- *Separação de responsabilidades* – Cada *fonte contextual* recolhe os dados e processa-os independentemente de onde estes irão ser utilizados, ou do funcionamento de outros sensores.

- *Interpretação contextual* – As abstrações em camadas de contexto são transparentes, tanto para o utilizador como para o programador, uma vez que os dados são transmitidos e distribuídos em todas as fases do ciclo de processamento, se assim for entendido.
- *Identificação de recursos* – o *gestor sensorial* possui um registo de todos os sensores disponíveis no equipamento e da forma de os consultar.
- *Comunicação transparente e distribuída* – o modelo proposto prevê a utilização de comunicação transparente de várias formas, com pedidos pontuais, por notificação direta, ou por disseminação.
- *Disponibilidade da aquisição de contexto constante* – as *fontes contextuais* funcionam independentemente das aplicações que os consomem.
- *Armazenamento de contexto* – o *gestor contextual* é responsável pelo arquivo e privacidade dos dados arquivados.

O ADAPT, pretende ainda dar resposta a alguns dos desafios, como o consumo energético, que é algo que pode limitar, severamente, as capacidades do sistema. De forma a dar resposta a esta problemática, o ADAPT, propõe utilizar uma gestão inteligente de recursos, não só através da gestão automática de frequência e dimensão temporal dos dados recebidos dos sensores, mas também, através da utilização de técnicas avançadas de monitorização e gestão energética eficiente.

4.4. Influências no modelo contextual antecipatório

A expressão “*de pé sobre ombros de gigantes*”, é comum no meio científico, pois todas as atividades de investigação deixam sempre um pouco de inspiração para futuros investigadores, que ambicionem explorar o desafio do desconhecido. Por este motivo e pela capacidade de

aprender com os sucessos e insucessos de experiências realizadas por outros investigadores, todos os projetos apresentados no Capítulo 3 exerceram, de certa forma, alguma influência no desenho e planeamento do modelo ADAPT, influência essa que teve uma importância variável, desde a consideração da temática ou abordagem para solucionar o problema, até aos casos com maior relevância, incluindo a orientação do modelo no mesmo sentido dessa proposta.

Ao nível arquitetural, existiram várias propostas que influenciaram o planeamento, como o CORTEX (Biegel & Cahill, 2004), que serviu de inspiração para a implementação de um protocolo de comunicação baseado no modelo de Evento-Condição-Ação, que permite tornar a aplicação reativa e preditiva ao contexto. Outras propostas permitiram dar alguma resposta ao nível das preocupações energéticas, como o SociableSense (Rachuri et al., 2010; Musolesi et al., 2011), com a sua técnica de amostragem variável.

A proposta CenceMe (Miluzzo et al., 2008), apresenta uma abordagem ao nível da delegação de contextos mais complexos e computacionalmente mais exigentes para serviços externos, onde esteja disponível maior capacidade de processamento. Complementarmente, através do conceito de configuração remota, proposto por SystemSens, (Falaki et al., 2011), foi possível aumentar ou diminuir o nível de detalhe da aquisição de partilha de dados para o serviço de processamento externo, sem ser necessário a reinstalação ou atualização da aplicação.

Ao nível do suporte sensorial, as soluções que mais tiveram influência foram as propostas OpenDataKit (Brunette et al., 2012), e AWARE (Ferreira et al., 2015), que apresentaram as vantagens existentes ao nível do desenvolvimento de uma camada de abstração sensorial e na capacidade de integração de novos sensores baseados numa arquitetura extensível com uma interface comum.

Ao nível do utilizador, foram também tomadas algumas decisões que tiveram origem: no sistema ContextStudio (Korpipää et al., 2004), nomeadamente ao nível da avaliação do utilizador sobre a inferência contextual realizada; no AWARE (Ferreira et al., 2015), ao nível da capacidade de controlo do tipo e configuração dos sensores a serem utilizados pelas aplicações; e no

Ginger.io (Ginger Inc, 2016), ao nível da abordagem, para a sensorização baseada no comportamento do utilizador e da representação gráfica desses mesmos dados ao utilizador.

Ao nível da privacidade, anonimização, exportação e partilha da informação contextual do utilizador, o ADAPT beneficiou das ideias e abordagens propostas por: “ContextPhone (Raento et al., 2005), AWARENESS (van Sinderen et al., 2006), MyExperience (Froehlich et al., 2007), Funf (Aharony et al., 2011), Ohmage (Tangmunarunkit et al., 2015), Empath (Dickerson et al., 2011), SociableSense (Rachuri et al., 2010; Musolesi et al., 2011)”.

4.5. Considerações finais do capítulo

Neste capítulo foi apresentado, um enquadramento dos sistemas *middleware* sensíveis ao contexto, apresentando as suas características e vantagens e apresentando, em seguida, alguns dos principais obstáculos ao desenvolvimento desses sistemas. Por fim, apresentou-se o nosso modelo conceptual de um *middleware* sensível ao contexto, ADAPT, modelo de adaptação dinâmica e sensível ao contexto de interfaces móveis em ambientes ubíquos onde se descreve, de uma forma introdutória, o modelo de adaptação contextual para a construção de aplicações móveis antecipatórias.

5. ARQUITETURA

Neste capítulo é apresentada uma arquitetura de suporte ao modelo contextual antecipatório ADAPT, desenvolvida para validar o modelo proposto. Este capítulo inicia-se com a apresentação da vista global do sistema, sendo apresentado de forma genérica e, progressivamente pormenorizados, cada um dos seus componentes, até ao nível da sua implementação.

5.1. Enquadramento

O modelo conceptual antecipatório, apresentado no Capítulo 4, define um modelo baseado na intenção de construir uma camada que permita a abstração de todos os sensores disponíveis e a utilização de contexto de forma uniformizada. O domínio de aplicação deste modelo centra-se nas necessidades do utilizador, ao nível da realização das suas ações, no momento presente, mas pode, no entanto, ser expandido a outros domínios de aplicação, inclusive a inferência de necessidades futuras. A aplicação deste modelo, permite que a utilização de contexto e a respetiva adaptação contextual, viabilizem o desenvolvimento de aplicações mais eficientes, que apoiem o utilizador consoante as suas necessidades.

No sentido de validar o modelo proposto, foi desenvolvida uma arquitetura de software que realiza o modelo conceptual antecipatório. A arquitetura proposta neste capítulo serve de base de suporte à implementação do caso de uso e testes realizados no Capítulo 6. O principal objetivo desta arquitetura é a implementação de um *middleware*, que implemente a sensorização contextual, de forma a permitir a adaptação dinâmica e antecipatória para aplicações móveis. Assim, a arquitetura deve assegurar:

- O processamento dos diversos sensores com base em regras e disponibilizar os dados de forma estruturada, independente da origem.
- Utilização e distribuição da informação sensorial, para avaliação e inferência do contexto do utilizador e para que múltiplas aplicações possam beneficiar do seu uso.
- Distribuição da inferência contextual em diversos níveis de processamento, de forma a permitir a utilização deste *middleware*, quando apenas se pretende o seu uso como uma camada de abstração dos sensores físicos.

Este capítulo inicia-se com a apresentação da vista global do sistema, descrevendo-se os seus principais componentes. Posteriormente, são expostos cada um dos seus componentes,

descritos em termos do seu funcionamento lógico, das suas funcionalidades e da sua implementação, detalhando em maior pormenor o seu funcionamento.

5.2. Vista global da arquitetura

A arquitetura de suporte ao modelo conceptual antecipatório para instrumentação de dispositivos móveis, desenvolvido no âmbito desta tese tem, por base, o modelo conceptual apresentado na Figura 5-1. Do ponto de vista funcional, pretende-se que o sistema funcione como uma camada de abstração do processamento dos sensores permitindo, assim, a redução da complexidade na interação da camada aplicacional com os sensores.

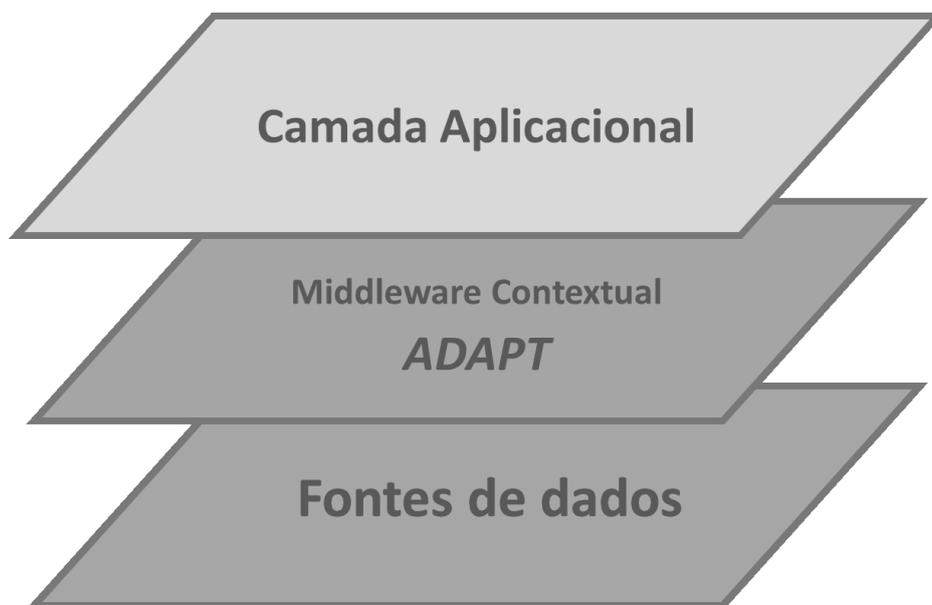


FIGURA 5-1 – MODELO CONCEPTUAL DE SUPORTE AO MODELO CONTEXTUAL ANTECIPATÓRIO

O ADAPT, tem como principal objetivo a redução da complexidade na utilização de sensores para inferência contextual em aplicações, aumentando a reutilização, através da criação de uma camada de abstração, sobre os sensores e fontes de informação. As fontes de dados são, por definição, um conjunto de múltiplos sensores que podem ser físicos, lógicos, ou mesmo interação direta com o utilizador.

O ADAPT interage com as diversas fontes de contexto e com a aplicação de técnicas de processamento, interpreta e infere o contexto e as necessidades do utilizador. Além disso, o ADAPT tem como principal objetivo e fundamento, a partilha e reutilização dessa informação com outras aplicações e serviços. O ADAPT tem como principal objetivo a redução da complexidade na utilização de sensores para inferência contextual em aplicações, aumentando a reutilização através da criação de uma camada de abstração sobre os sensores e fontes de informação

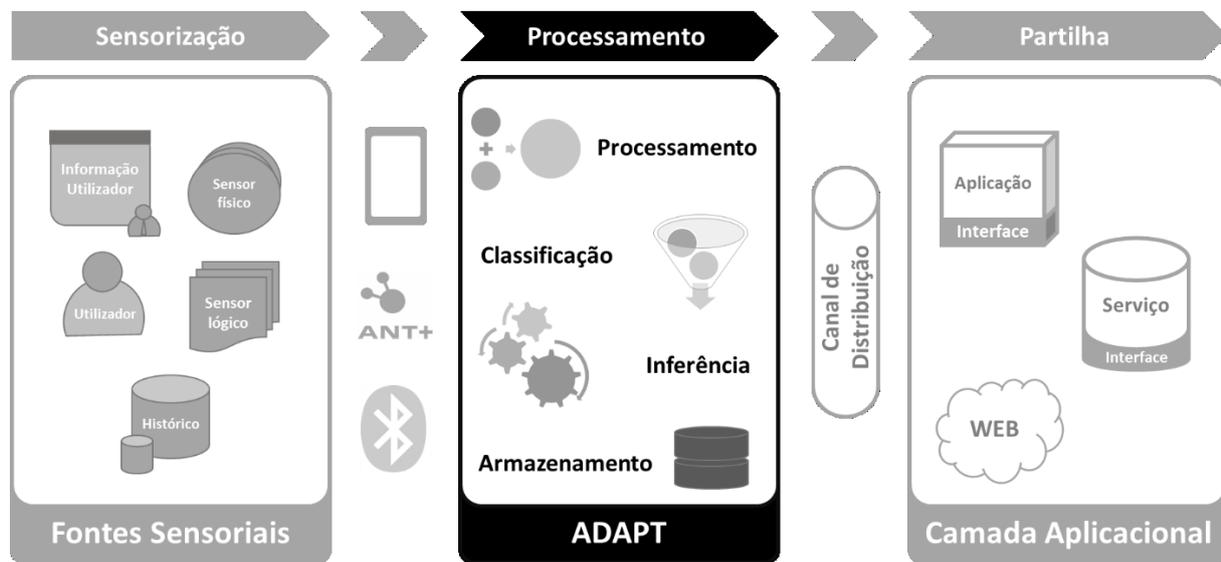


FIGURA 5-2 – VISTA GLOBAL DA ARQUITETURA DO ADAPT

Uma das características do ADAPT é o funcionamento modular e dinâmico, isto é, através da utilização do conceito de canal de distribuição comum, permite que o sistema utilize dinamicamente os sensores que pretende, de forma independente e isolada de outros.

Esta abordagem permite-nos isolar o processamento dos sensores e do respetivo contexto, aumentando a estabilidade do sistema uma vez que, mesmo que uma das fontes contextuais falhe, não se compromete a estabilidade da aplicação. Outro dos motivos que nos levou a selecionar esta abordagem, é a diversidade de características e disponibilidade de sensores nos dispositivos, pelo que é mais eficiente que os sensores sejam utilizados dinamicamente e consoante a sua disponibilidade e necessidade.

5.3. Vista funcional da arquitetura

A principal característica do ADAPT é a sua visão de funcionamento desacoplado, onde as fontes de contexto, independentemente da sua categoria contextual, funcionam de forma independente e isolada. Assim, é possível criar uma plataforma extensível, resiliente e com tolerância a falhas uma vez que, por definição, a implementação ao nível do consumo destes eventos obriga à prevenção de potenciais erros, ou mesmo, à indisponibilidade da fonte de contexto. Por outro lado, a arquitetura modular, em que cada fonte contextual é responsável por apenas uma e uma só função, promove a capacidade de terminar e recuperar o processo de forma limpa e sem prejudicar o funcionamento de outras.

Esta decisão arquitetural é inspirada no conceito de micro serviços, uma arquitetura de desenvolvimento de software que propõe a decomposição das responsabilidades do sistema em unidades isoladas e independentes, onde cada unidade tem apenas uma responsabilidade, que deve realizar com sucesso. Sendo uma evolução natural da arquitetura orientada a serviços, desenhada segundo as capacidades do domínio onde é aplicado, os micro serviços implementam muitos dos conceitos propostos por Evans (2004) para o desenvolvimento de uma aplicação composta por componentes independentes.

Os micro serviços apresentam-se, assim, como uma abordagem para o desenvolvimento de sistemas, constituídos por um conjunto de pequenos serviços, cada um contido no seu próprio processo, interagindo com outros, tipicamente através de uma API HTTP (Fowler & Lewis, 2014; Fowler, 2015; Newman, 2015). Os micro serviços caracterizam-se por possuir uma interface publicamente definida, sendo totalmente autónomos, isolados e autogeridos, isto é, eles são responsáveis pela gestão do seu ciclo de vida (Newman, 2015). Por este motivo, a alteração da implementação de um microserviço não produz impacto no sistema, pois todos os elementos, no ecossistema de micro serviços, são desacoplados e com elevados níveis de coesão, uma vez que comunicam através de interfaces definidas, que permanecem imutáveis para o microserviço.

Esta arquitetura requer, no entanto, uma camada mínima de administração e gestão de serviço, que pode variar de complexidade e funcionalidades mas tem, no mínimo, a responsabilidade da gestão de disponibilidade e manutenção da saúde dos micro serviços, entre outras (Pahl & Lee, 2015). Devido à interface de comunicação que os micro serviços possuem, estes podem ser desenhados em várias linguagens de programação e utilizar diferentes formas de armazenamento, o que os torna bastante interessantes, pois permite utilizar as melhores tecnologias para dar resposta a cada um dos desafios do domínio da aplicação.

A aplicação desta proposta arquitetural, de serviços isolados distribuídos no conceito de sensibilidade ao contexto, de forma a permitir beneficiar das vantagens de reutilização e simplificação e aceleração do processo de desenvolvimento, não é totalmente inovadora. Bak et al. (2015), sugeriram a utilização desta arquitetura em micro serviços, implementados na nuvem, que permitiam inferir e processar a informação de contexto do utilizador nomeadamente, a localização ou as preferências do mesmo, para lhe disponibilizar cupões de desconto. Esta proposta tem, no entanto, um requisito estrutural, a necessidade de ligação permanente à internet durante o seu funcionamento. No ADAPT temos como inovação, a implementação do conceito simplificado de micro serviços, implementado ao nível do dispositivo e removendo, assim, a necessidade da dependência da ligação permanente à internet, mas mantendo os benefícios que este tipo de arquitetura proporciona.

Numa implementação tradicional, mais inspirada num processo monolítico, a aplicação é responsável por tudo e, por esse mesmo motivo, apresenta menor fiabilidade, pois o erro grave de um dos sensores pode levar a que a aplicação termine inesperadamente. Assim, no ADAPT, ao invés de possuímos um único processo complexo para o processamento dos dados dos diversos sensores de informação contextual, possuímos sim, um conjunto de módulos sensoriais isolados e independentes, que comunicam uns com os outros de forma distribuída e assíncrona, aumentando a estabilidade e fiabilidade de todo o sistema.

O ADAPT, representado na Figura 5-3, é constituído por um conjunto de módulos lógicos e funcionais que representa, de forma estrutural, as nossas decisões de arquitetura. O sistema é

constituído, de forma geral, por um canal de comunicação comum e dois módulos principais, a gestão contextual e a fonte contextual.

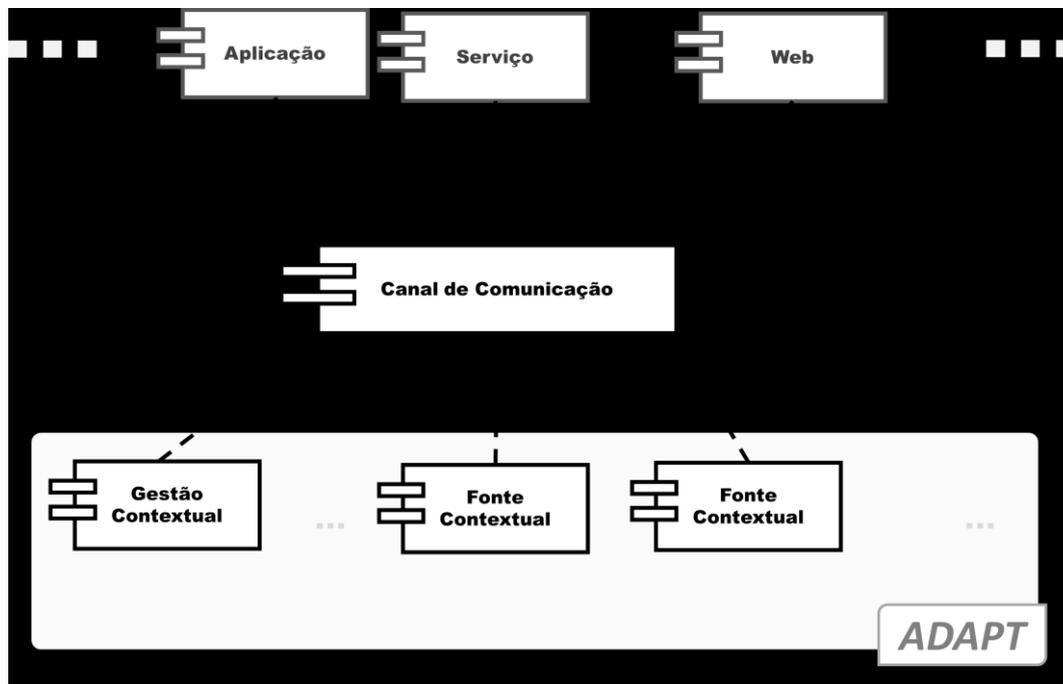


FIGURA 5-3 – VISTA DE COMPONENTES DO ADAPT

O ADAPT inicia-se pela **gestão contextual** que possui duas funcionalidades principais: o **catálogo sensorial** e a **gestão sensorial** em si:

O **Catálogo Sensorial** é um módulo de registo que permite às aplicações que interagem com o ADAPT, ter conhecimento de quais as fontes contextuais disponíveis e a forma de interação com as mesmas. Após a seleção das fontes contextuais pretendidas, por parte da aplicação consumidora com o módulo de gestão sensorial, o ADAPT irá funcionar de forma automática, com cada fonte contextual a funcionar de forma isolada, com base nos requisitos definidos pelo consumidor do serviço.

A **Gestão Sensorial**, é um módulo responsável pela administração das fontes sensoriais, e pela respetiva gestão energética onde, através da monitorização da capacidade de energia restante no dispositivo ao longo do funcionamento do sistema, irá proceder à gestão otimizada dos sensores, podendo alterar dinamicamente (aumentar ou diminuir) a frequência de atualização e/ou aquisição dos sensores, de forma automática.

O **canal de comunicação / distribuição** é um canal comum aos micro serviços, onde todos expõem as suas interfaces de comunicação permitindo, assim, receber e partilhar informação com outros serviços e aplicações que o requisitem. Este canal é o mais comum possível, de forma a permitir reduzir a necessidade de acrescentar mais um requisito ao desenvolvimento de aplicações que consomem informação contextual.

As **Fontes Contextuais**, são os micro serviços responsáveis por adquirir os dados dos sensores e, através de um conjunto de processos, obter a informação contextual que irá ser utilizada pela aplicação que a requisitou, podendo ser distribuída em diversas fases do processo de construção da informação contextual.

5.3.1. Gestão Contextual

A gestão contextual é um módulo responsável por duas funcionalidades principais, o catálogo sensorial e a gestão sensorial (Figura 5-4). Estes serviços são fundamentais para o funcionamento do ADAPT pois, é através deles que toda a interação com as fontes contextuais é construída.

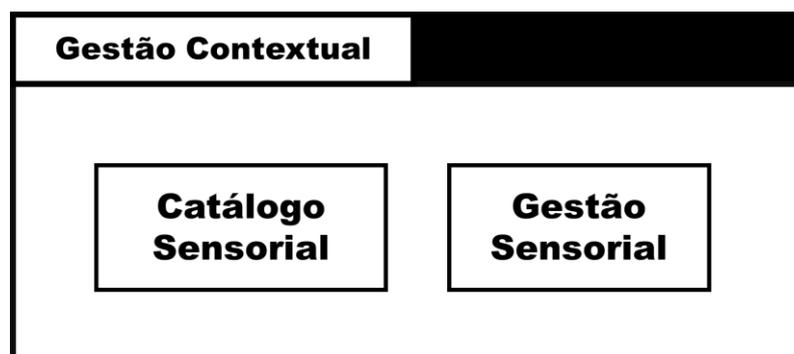


FIGURA 5-4 – ESTRUTURA DA GESTÃO CONTEXTUAL

Na utilização de informação contextual no ADAPT, existe um processo de interação entre os diferentes módulos para o consumo dessa informação, sendo esse fluxo representado pela Figura 5-5.

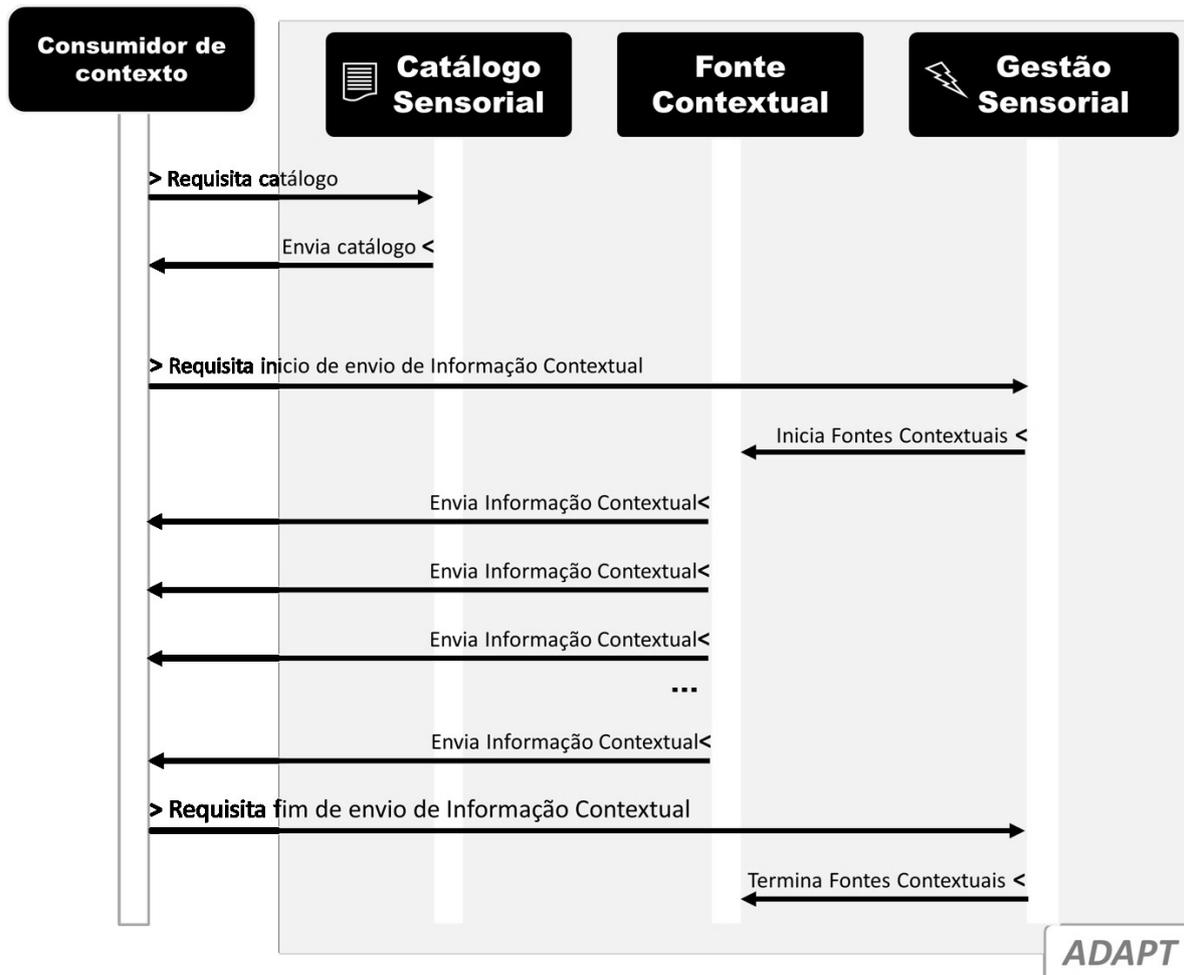


FIGURA 5-5 – VISTA DE SEQUÊNCIA DE PEDIDO DA INFORMAÇÃO CONTEXTUAL

A primeira interação das aplicações consumidoras de informação contextual, na fase inicial do processo será com o Catálogo Sensorial, uma vez que estas necessitam de saber quais as capacidades contextuais disponíveis no dispositivo.

Catálogo sensorial

O Catálogo Sensorial é um microserviço que contém registo interno cuja responsabilidade é a de possuir o conhecimento de quais as fontes contextuais disponíveis no dispositivo, ou possíveis de instalar, e a forma de interação com as mesmas, nomeadamente as suas características ou restrições. Para corresponder a esse objetivo, é necessária a existência de uma estrutura de dados (Figura 5-6) que contenha toda a informação necessária para tomar essas decisões.

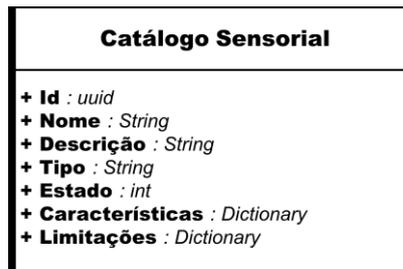


FIGURA 5-6 – ESTRUTURA DE DADOS DO CATÁLOGO SENSORIAL

Os componentes dessa lista devem permitir conter a seguinte informação:

- *Id* – um identificador único da fonte sensorial, que irá ser utilizado para identificar inequivocamente essa fonte e, assim, poder requisitar o início ou o fim dos dados.
- *Nome* – nome do identificador, normalmente uma versão resumida das capacidades do sensor (ex. sensor de movimento), que tem como objetivo permitir informar o utilizador de forma rápida sobre as capacidades da fonte contextual.
- *Descrição* – descrição detalhada sobre as capacidades do sensor, nomeadamente, o tipo de informação que esta fonte contextual pode possuir, de forma a informar o utilizador sobre as funcionalidades e, assim, permitir que este possa controlar melhor a utilização das fontes contextuais.
- *Tipo* – o tipo de fonte contextual é relativo à categoria de informação que o sensor apresenta, ou seja, os tipos de dados contextuais, por ex. localização externa / interna, identificador de atividade, horário/agenda do utilizador, entre outros.
- *Estado* – o estado tem com objetivo representar o estado da fonte sensorial, nomeadamente, se esta se encontra disponível e em funcionamento normal, em funcionamento restrito (por seleção do utilizador, ou questões

energéticas), ou mesmo indisponível, no caso de ter sido desativado pelo utilizador.

- *Características* – conjunto de características que permitem personalizar o modo de funcionamento da fonte contextual, como, Frequência de Atualização (por defeito, intervalo incremental), Interface de Comunicação (genérico, específico), Tipo de Processamento (local/remoto/hibrido), Níveis de Distribuição (processados, categorizados, inferidos).
- *Limitações* – conjunto de informações relativas ao funcionamento em modo limitado, possui informações como Frequência de Atualização, Níveis de distribuição ativos.

Após receber o catálogo com a informação das fontes sensoriais, a aplicação é responsável por fazer uma seleção das fontes contextuais que pretende receber, com base nas características das existentes e, em, seguida realizar o pedido de contexto ao serviço de gestão sensorial.

Gestão sensorial

A gestão sensorial (Figura 5-7) é um microserviço responsável pela administração e controlo das fontes sensoriais, nomeadamente, de gerir quais as fontes de contexto a iniciar, ou terminar e a gestão das taxas de frequência, consoante as necessidades de informação contextual ou as capacidades energéticas do dispositivo.

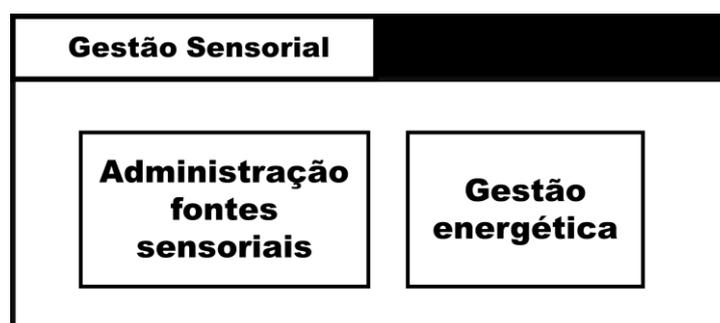


FIGURA 5-7 – ESTRUTURA DO MICROSERVIÇO GESTÃO SENSORIAL

O seu processo de funcionamento, inicia-se tal como descrito na Figura 5-5, apresentada anteriormente (pág. 106), com o pedido por parte do consumidor de contexto do sensor que pretende ativar (*id*) e o conjunto de características com que pretende que este seja iniciado. Este pedido irá ser processado pela administração de fontes sensoriais, que possui a referência de todos os serviços que foram iniciados e, caso ainda não exista nenhuma fonte contextual, que corresponda às necessidades requisitadas pelo utilizador, irá requisitar o início de uma.

Na fase inicial, a fonte contextual irá autoconfigurar-se e funcionar de forma independente do resto do sistema sendo este responsável, no entanto, por realizar o registo no módulo de gestão sensorial. Este processo de registo é necessário devido à arquitetura promover o uso de micro serviços isolados e heterogéneos para a sua implementação. Mais detalhe acerca da sua implementação poderá ser consultado na secção seguinte.

Em paralelo a este processo, será iniciado também o módulo de gestão energética, que é responsável por reduzir o impacto que o processamento contextual possui no consumo energético.

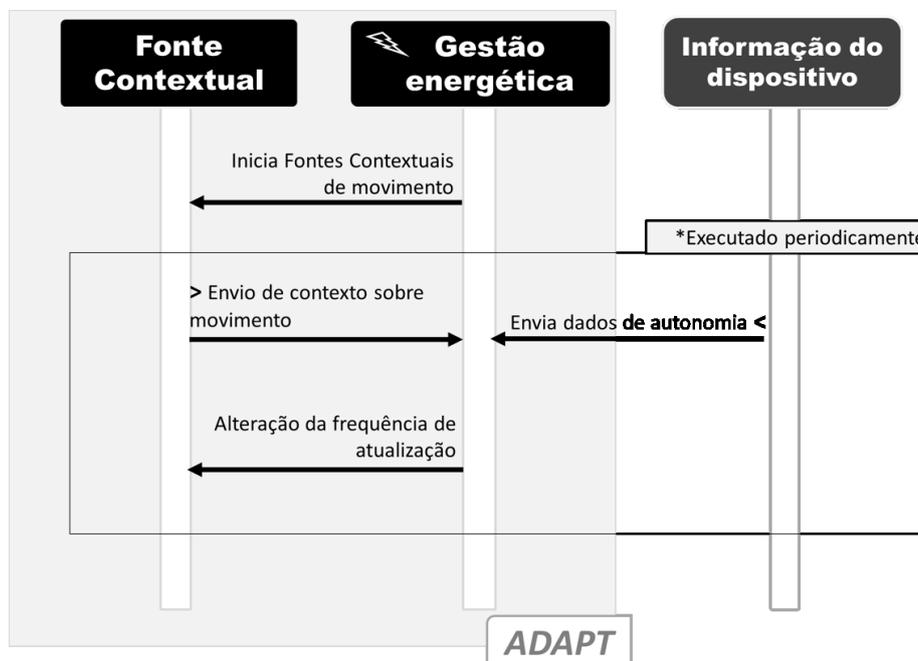


FIGURA 5-8 – VISTA DE SEQUÊNCIA DO MODELO DE GESTÃO ENERGÉTICA

Tal como representado na Figura 5-8, o processo de gestão energética, é ele próprio um consumidor de informação contextual, uma vez que o contexto de movimento do utilizador afeta as necessidades ao nível da frequência de atualização, por exemplo, ao nível do contexto de localização. Assim, o módulo irá receber periodicamente dois tipos de informação contextual: a capacidade de autonomia restante do dispositivo e o contexto movimentação do utilizador. Com base nesta informação, irá tomar decisões ao nível da variação da frequência de atualização.

De uma forma mais específica, por exemplo, para uma fonte contextual de localização mista (interna e externa), o módulo irá reduzir a utilização e frequência de atualização do GPS apenas em situações em que é realmente necessário e útil. Uma das formas em que ele o faz é, por exemplo, variando o tempo entre recolhas, tendo como base a deteção de situações em que o utilizador não está em movimento logo, a alteração da posição não existe. Em situações mais complexas, quando é detetado que o utilizador se encontra dentro de um edifício, este módulo irá desativar a recolha de localização via GPS, uma vez que dentro de edifícios, as coordenadas obtidas não são confiáveis.

De forma a manter a capacidade de gestão e administração dos micro serviços é necessário que esta classe possua a capacidade de manter uma lista atualizada com os serviços disponíveis e como comunicar com os eles. Uma das abordagens mais eficazes para este problema é a utilização de um *Service Locator*, com uma abordagem *Singleton* (Fowler, 2004).

Exemplo de Implementação do ADAPTServiceLocator

```
public interface IADAPTContextServiceLocator
{
    T GetService<T>();
}
class ADAPTServiceLocator : IADAPTContextServiceLocator
{
    private IDictionary<Type, Type> servicesTypeList;
    private static readonly object theLock = new Object();
    private readonly IDictionary<Type, object> runningServices;
    private static IServiceLocator instance;
    public T GetService<T>()
    {
        if (this.runningServices.ContainsKey(typeof(T)))
        {
            return (T)this.runningServices[typeof(T)];
        }
        else
        {
```

```

        try
        {
            // usar reflexão para invocar o método no serviço
            var constructor = servicesTypeList [typeof(T)].GetConstructor(new Type[0]);
            T service = (T)constructor.Invoke(null);
            runningServices.Add(typeof(T), service);
            return service;
        }
        catch (KeyNotFoundException)
        {
            throw new ApplicationException("The requested service is not registered");
        }
    }
}
public static IServiceLocator Instance
{
    get
    {
        lock (theLock) // thread safety
        {
            if (instance == null)
            {
                instance = new ADAPTServiceLocator();
            }
        }
        return instance;
    }
}
}

```

O exemplo apresentado na especifica o **ADAPTServiceLocator**, uma implementação de um *service locator* com uma abordagem *singleton*. Na sua constituição o **ADAPTServiceLocator** possui 4 objetos:

- private IDictionary<Type, Type> servicesTypeList; – que é constituído pela lista de micro serviços disponíveis no dispositivo. Este objeto é preenchido ao iniciar do ADAPT, e sempre que existem alterações por parte do utilizador na configuração, sendo inicializado da seguinte forma:

```

//Regista na lista de serviços disponíveis provedor de localização interna
servicesTypeList.Add(typeof(IServiceInternalLocation), (ServiceInternalLocation));

```

- private readonly IDictionary<Type, object> runningServices; - lista de referência para serviços que estão a ser executados em um determinado momento;

- `private static IServiceLocator instance;` - A instância `ADAPTServiceLocator` propriamente dita, e uma vez o seu construtor é privado, apenas é possível aceder-lhe através desta propriedade, sendo este *service locator* utilizado da seguinte forma:

```
//obtem o produtor de localização interna
var internalLocationSrv =
ADAPTServiceLocator.Instance.GetService<IServiceInternalLocation>();
```

- `private static readonly object theLock = new Object();` - um objeto utilizado como bloqueador para evitar os problemas de concorrência em multi-threads (Skeet, 2014).

Com a abordagem apresentada é assim possível que o módulo de gestão sensorial realize a administração dos micro serviços de fontes contextuais, uma vez que cada um dos serviços possui uma interface e implementação que lhe permite comunicar. Por exemplo, no caso do *IServiceInternalLocation* e *ServiceInternalLocation*, uma vez que estes poderiam ser implementados noutra linguagem, que não a que implementa o *ServiceLocator*, existe a necessidade de implementar uma interface para comunicar com o mesmo, nomeadamente, iniciar e administrar o processo.

Exemplo de Implementação do ServiceInternalLocation

```
// Interface base implementada por todos os módulos de comunicação
public interface IBaseService
{
    StartService();
    UpdateParameters(T);
    TerminateService();
}
// Interface base implementada pelos módulos do tipo Contexto Localização Interna
public interface IServiceInternalLocation
{
    GetCurrentLocation();
}
// Esta classe envia mensagem através do canal de comunicação para o processo iniciar.
// e o processo como irá subscrever eventos desse tipo, será automaticamente iniciado
internal class ServiceInternalLocation : IServiceInternalLocation, IBaseService
{
    public void StartService()
    {
        ChannelBroker.SendEvent(ServiceInternalLocation.Start);
    }
    public void UpdateParameters(object[] parms)
    {
        ChannelBroker.SendEvent(ServiceInternalLocation.UpdateParameters, parms);
    }
}
```

```
public void TerminateService()
{
    ChannelBroker.SendEvent(ServiceInternalLocation.Stop);
}
public void GetCurrentLocation()
{
    ChannelBroker.SendEvent(ServiceInternalLocation.GetCurrentLocation);
}
}
```

No exemplo apresentado em cima é possível verificar um exemplo da implementação do modelo de comunicação proposto na implementação do *ADAPTServiceLocator* uma vez que, na nossa visão, os serviços são unidades isoladas, independentes e tecnologicamente heterogêneos, pelo que existe a necessidade de um canal de comunicação para permitir a comunicação e interação entre serviços.

5.3.2. Canal de comunicação

Na nossa proposta arquitetural é referido, em diversas ocasiões, a existência de um canal de comunicação que permite, aos diferentes componentes da ADAPT, comunicar entre si e também entre os micro serviços e as aplicações consumidoras. De uma forma geral, o canal de comunicação é a estrada que liga todos os destinos. Até aqui o canal de comunicação foi sempre referido de forma abstrata, devido à sua simplicidade, apesar da sua importância fundamental para o sistema pelo que é, por isso, importante especificá-lo.

O canal de comunicação proposto é um sistema, independente da arquitetura ADAPT, que implementa o padrão de desenvolvimento *Message Broker* com capacidade *publish-subscribe* (Hohpe & Woolf, 2004) que tem como principal funcionalidade receber e entregar mensagens aos seus destinatários, isto é, um consumidor regista neste serviço o interesse de receber mensagens de um determinado tipo, e o broker é responsável por lhe entregar todas as mensagens que sejam registadas, posteriormente, nesse mesmo canal. Tipicamente, as plataformas móveis possuem um sistema interno de comunicação inter-aplicacional que implementa, de alguma forma, as necessidades de um *broker publish-subscribe*, como é o caso do Android, com o seu *BroadcastChannel*.

No entanto, existem cenários onde os brokers existentes não dão resposta suficiente às necessidades de um determinado problema, como comunicação interna e externa via internet e, é aí que existem soluções mais específicas, como o *MQTT – Message Queue Telemetry Transport*, que é dedicado à criação de um canal de comunicação para informação de telemetria, que também se aplica ao domínio do nosso problema. *O MQTT* é um standard ISO e OASIS que especifica a implementação de um protocolo de comunicação mais avançado sobre o protocolo TCP/IP, o que permite abranger um maior número de cenários, como suportar comunicação local e remota, de forma transparente para as aplicações consumidoras. Um exemplo destes sistemas é o *MOSQUITTO* (<https://mosquitto.org/>), este selecionado por possuir uma implementação disponível em diversas plataformas Windows, Linux, e também móveis como o android.

No caso do Android, este conceito é implementado pelo *Broadcast* (Google inc, 2017), um componente da plataforma Android que permite às aplicações comunicar e transferir informação. Este canal, implementa o conceito de produtor-subscritor, o que permite que os eventos sejam enviados para as aplicações que necessitam deles.

As subscrições podem ser realizadas de duas formas, via configuração estática através do ficheiro *AndroidManifest.xml* ou dinamicamente através do método de registo *Context.registerReceiver()* e implementando uma classe para receber a informação destes eventos. Um requisito importante é que estes eventos sejam assíncronos, pois uma aplicação não sabe, necessariamente, quando irá receber o evento, sabe apenas que o irá receber.

Exemplo de registo dinâmico de receção de eventos

```
BroadcastReceiver br = new MyBroadcastReceiver();
IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
intentFilter.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED);
this.registerReceiver(br, filter);
```

Este canal pode também ser utilizado para enviar informação, podendo variar entre dois tipos, ordenado ou aleatório, sendo que no primeiro caso, a prioridade é definida pelo consumidor. O android irá priorizar a entrega dos eventos primeiro a essas aplicações, pela ordem do valor da variável *android:priority*. De referir que o método ordenado tem uma performance

inferior ao método aleatório, mas possui a vantagem de o evento poder ser cancelado, enquanto que o modo normal, apesar de mais eficiente, não possui esta capacidade.

Exemplo de envio de um evento para o Canal de comunicação

```
Intent intent = new Intent();
intent.setAction("com.example.broadcast.MY_NOTIFICATION");
intent.putExtra("data", "Dados a Enviar!");
sendBroadcast(intent);
```

No caso do ADAPT, como o controlo do fluxo de funcionamento das aplicações é realizado pela publicação/consumo de eventos através do canal de comunicação, foi necessário a especificação de um conjunto de interfaces, nomeadamente ao nível do registo da necessidade de contexto.

Na Tabela 5-1, é apresentado uma descrição do tipo de eventos que cada uma das fontes contextuais irá subscrever e, com base nelas, realizar operações pretendidas.

TABELA 5-1 – EXEMPLO DE EVENTOS PARA CADA TIPO DE FONTE CONTEXTUAL

ADAPT.[CONTEXT_SOURCE_NAME].START	EVENTO QUE INDICA A NECESSIDADE DE INÍCIO DE PUBLICAÇÃO DE CONTEXTO POR PARTE DESSA FONTE CONTEXTUAL
ADAPT.[CONTEXT_SOURCE_NAME].STOP	EVENTO QUE INDICA PARA A FONTE CONTEXTUAL INTERROMPER A PUBLICAÇÃO DE CONTEXTO. NOTA, UMA VEZ QUE É POSSÍVEL VÁRIAS APLICAÇÕES CONSUMIR A MESMA INFORMAÇÃO, A FONTE CONTEXTUAL SÓ IRA TERMINAR QUANDO RECEBER O MESMO NÚMERO DE INTERRUPÇÕES COMO DE INICIAÇÃO
ADAPT.[CONTEXT_SOURCE_NAME].LOWBATTERY	EVENTO QUE INDICA À FONTE CONTEXTUAL QUE O DISPOSITIVO SE ENCONTRA COM AUTONOMIA REDUZIDA, PELO QUE TEM QUE DIMINUIR O CONSUMO EFETUADO, NOMEADAMENTE DIMINUINDO A TAXA DE ATUALIZAÇÃO

Existem outros serviços que estendem esta lista, ao implementar outros eventos mas, no entanto, irá sempre possuir no mínimo este conjunto de eventos de forma a permitir a funcionalidade base.

5.3.3. Fonte contextual

A fonte contextual é o elemento principal do ADAPT, sendo responsável pela aquisição dos dados dos diversos sensores e, após o seu processamento, obter um conjunto de informação contextual que possa ser utilizada por outras aplicações para adaptar o seu comportamento às necessidades do utilizador. Este processo, como referido anteriormente, na visão de uma arquitetura baseada em micro serviços, são processos independentes, isolados, heterógenos, e tem à sua responsabilidade a gestão do seu ciclo de vida, desde que são invocados, passando pelas diversas interações e distribuição de contexto até por fim, ao seu encerramento quando não são mais necessárias.

Na Figura 5-9, é apresentado, em maior detalhe, o processo interno de funcionamento da Fonte Contextual, destacando-se o fluxo de transformação dos dados sensoriais, desde a sua aquisição, durante as diversas fases de distribuição ao longo do processamento, e o seu armazenamento para posterior reutilização, se necessário.

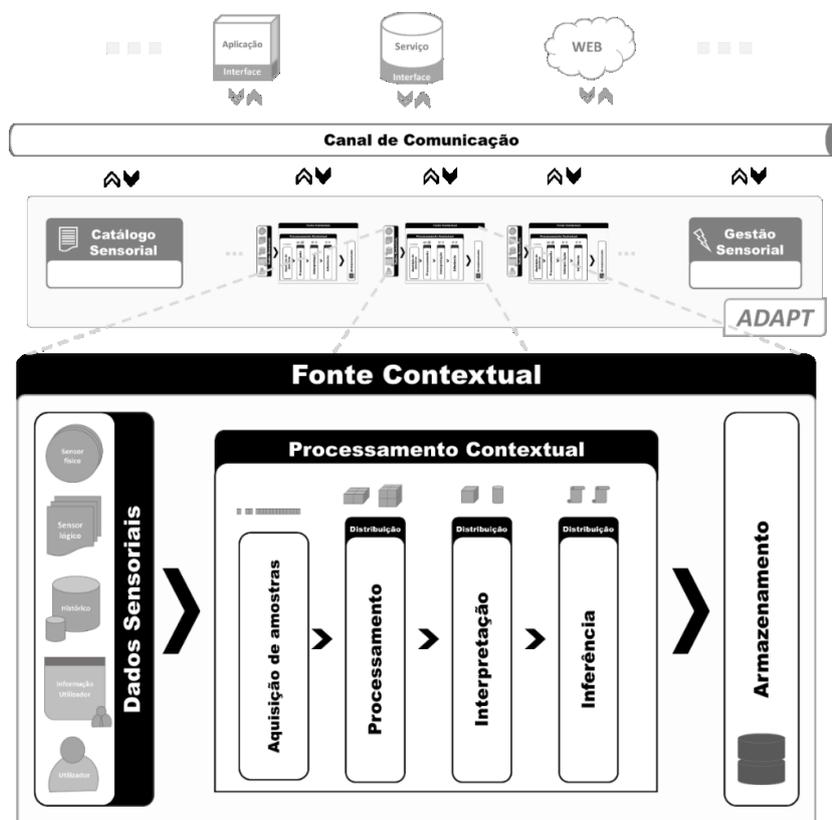


FIGURA 5-9 – PROCESSO INTERNO DE FUNCIONAMENTO DA FONTE CONTEXTUAL DO ADAPT

Após a invocação inicial do microserviço da fonte contextual, está a execução da sua função, iniciando o processo pela fase da sensorização, onde é realizada a **aquisição de amostras** dos dados sensoriais, com base nos sensores que essa fonte contextual necessita. Nesta fase, o processo realiza as amostragens dos sensores consoante as características de cada um dos sensores e a frequência permitida pela **Gestão Sensorial**, de forma a realizar o processamento e transformação dos dados.

Ao nível da origem dos dados, pretende-se que as fontes contextuais possam obter informação a partir de um conjunto diverso de sensores, entre os quais:

- *Informação do utilizador* – Informação relativa ao utilizador, incluindo dados pessoais, as suas preferências, a sua agenda ou horário, as suas tarefas, entre outras;
- *Sensores Físicos* – dados obtidos a partir de sensores de hardware, tais como localização por GPS, acelerómetro e giroscópio, temperatura, luminosidade, monitores cardíacos, contadores de passos, incluindo os sensores internos e externos do dispositivo;
- *Dados Lógicos* – sensores virtuais, que podem ser obtidos por processamento, agrupamento de sensores físicos e de fontes de outras características;
- *Respostas do Utilizador* – informação obtida diretamente por interação do utilizador, podendo ter o formato de avaliação em escalas de *likert*, por seleção de opção, entre outros.
- *Histórico anterior* – informação arquivada de ações e contextos processados anteriormente e registadas em arquivo.

Na fase de **Processamento**, após a aquisição dos dados, segue-se um conjunto de processos de transformação com vista à interpretação da informação contextual. O primeiro processo consiste num pré-processamento simples dos dados, com vista à sua uniformização, remoção de possíveis anomalias e agrupamento em unidade lógicas contextuais (ex. numa fonte de contextual de Localização do Utilizador, seriam reunidas todas as informações úteis para a

localização, tais como, informação GPS, WIFI, *Bluetooth Beacons*, entre outros). Após esta primeira fase de processamento, estes dados serão publicados no canal de distribuição, caso tenha sido requisitado sendo, em simultâneo, enviados para a fase seguinte do ciclo de processamento interno.

O segundo processo, a **Interpretação**, é onde será realizada a primeira fase de processamento com técnicas de classificação, categorização, interpretação e avaliação do contexto do utilizador. Esta fase tem como objetivo principal transmitir às aplicações a informação do contexto atual e das ações que o utilizador está a realizar. De forma semelhante à fase anterior, estes dados são transmitidos para o canal de distribuição comum e enviados, em simultâneo, para a fase seguinte - a fase de inferência.

Na fase de **Inferência**, caso exista, será realizado o processamento final dos dados. Nesta fase, pretende-se antecipar o contexto futuro do utilizador, nomeadamente, quais as futuras ações que este pretende realizar e quais as suas necessidades para a realização, com sucesso, dessas mesmas ações. Após a transmissão destes dados para os consumidores, os resultados de todas estas fases de processamento serão registados na base de dados, de forma a permitir a sua posterior reutilização.

Apesar de, na nossa implementação, se privilegiar o processamento local, a arquitetura proposta permite que qualquer uma das fases, de **processamento**, **interpretação** ou **inferência**, seja realizada remotamente, em parte ou na totalidade. Deste modo, dependendo da implementação da **Fonte Contextual**, é possível construir sensores locais, remotos, híbridos e remotos com tolerância à falha utilizando, para isso por exemplo, um modo local com capacidades mais reduzidas para situações em que não possua ligação à internet.

A camada de **armazenamento**, também opcional consoante as necessidades da fonte sensorial, têm como objetivo a persistência de informação resultante de processamentos anteriores para aplicação em situações futuras.

Pipeline de processamento contextual

Como é possível observar na Figura 5-10, as fases apresentadas são realizadas em sequência, usando o mesmo conjunto de dados, que irão sofrendo alterações à medida que são processados em cada uma das fases da pipeline de processamento contextual. Na Figura 5-10, são apresentadas, em detalhe, as principais fases do ciclo de vida da fonte contextual, que se inicia com a publicação do evento correspondente à requisição do processo, através de invocação assíncrona por parte do módulo de **Gestão Sensorial**. A atividade é iniciada após a respetiva receção.

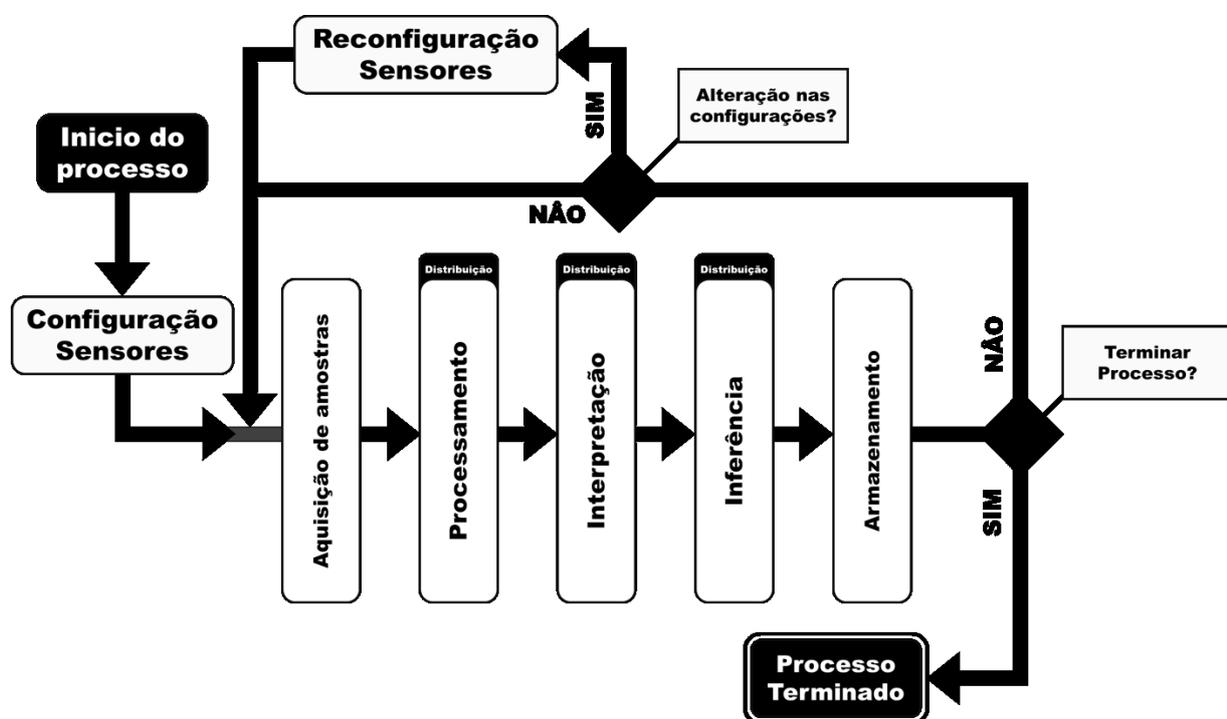


FIGURA 5-10 – PIPELINE SEQUENCIAL DE PROCESSAMENTO DA INFORMAÇÃO CONTEXTUAL

Após o início do processo, seguir-se-á a configuração dos sensores e dos subprocessos de aquisição de dados e, assim, dá-se início ao processamento da informação contextual, segundo o processo contextual. De forma a ser possível implementar este processo sequencial, é necessária a existência de um conjunto de estruturas de dados para dar suporte a esta sequência funcional, conforme ilustrado em (Figura 5-10), com o respetivo exemplo de implementação.

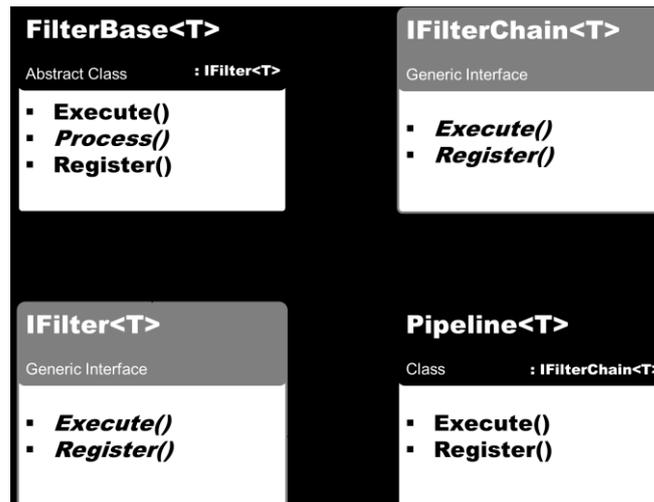


FIGURA 5-11 – MODELO DE CLASSES DA PIPELINE DE SUPORTE AO PROCESSAMENTO CONTEXTUAL

Exemplo de implementação do modelo de classes de suporte à pipeline

```

public interface IFilter<T>
{
    T Execute(T);
    void Register(IFilter<T>);
}

public abstract class FilterBase<T> : IFilter<T>
{
    private IFilter<T> _nextStep;
    protected abstract T Process(T inputStep);
    public T Execute(T inputStep)
    {
        T step = Process(inputStep);
        if(_nextStep != null)
        {
            step = _nextStep.Execute(step);
        }
        return val;
    }
    public void Register(IFilter<T> step)
    {
        if(_nextStep == null)
        {
            _nextStep = step;
        }
        else
        {
            _nextStep.Register(step);
        }
    }
}

public interface IFilterChain<T>
{
    void Execute(T);
    IFilterChain<T> Register(IFilter<T>);
}
  
```

```

public class Pipeline<T> : IFilterChain<T>
{
    private IFilter<T> _baseStep;

    public void Execute(T input)
    {
        _baseStep.Execute(input);
    }
    public IFilterChain<T> Register(IFilter<T> step)
    {
        if(_baseStep == null)
        {
            _baseStep = step;
        }
        else
        {
            _baseStep.Register(step);
        }
        return this;
    }
}

```

A principal vantagem da utilização da interface extra **IFilterChain<T>**, é a capacidade de permitir encadear os processos da pipeline, utilizando uma linguagem específica do domínio aplicacional interna (Fowler & Parsons, 2011). Assim, é possível, de uma forma simples, ao olhar para o código que implementa uma determinada pipeline, perceber imediatamente o processo sequencial bem como, quais são os processos que a suporta.

Exemplo de implementação da pipeline utilizando a interface fluente

```

var pipeline = new Pipeline<ContextObject>();
var ctxObj = new ContextObject();

//registar as diversas fases do processo por ordem de execução
pipeline.Register(new DataSampleAquisitionProcess())
    .Register(new DataCleanUpAnomalyDetectionProcess())
    .Register(new ContextInterpretationProcess())
    .Register(new ContextPredictionProcess())
    .Register(new DataStorageProcess())
    .Execute(ctxObj); //o processo irá iniciar-se pela ordem do registo.

```

Uma vez que se pretende que a pipeline seja executada em série para uma determinada amostra, o exemplo aqui apresentado é executado de forma síncrona mas, uma vez que toda a arquitetura é assíncrona, nada impede a realização de múltiplas pipelines em paralelo, se assim for necessário, uma vez que cada interação da pipeline é independente das restantes.

Após conclusão da interação da pipeline, o processo irá verificar se foi requisitada a conclusão do processo por parte da aplicação que o solicitou inicialmente. Em caso afirmativo, este processo irá concluir terminando a sua execução; em caso negativo, irá repetir mais uma vez a execução da pipeline sendo que, antes de começar, o microserviço verifica se existe alguma alteração das configurações de frequência de atualização, nomeadamente, das restrições impostas pelo módulo de gestão energética. Este ajuste da frequência de atualização é requisitado assincronamente pelo gestor energético e aplicado pela fonte contextual apenas após a conclusão do ciclo de execução da pipeline, apesar de poder receber a notificação antes.

Localização interna

A problemática da localização interna já foi introduzida no Capítulo 2 pelo que, nesta secção, iremos apresentar a implementação realizada com a técnica *Random Forest*, apresentado na secção 2.5.1 (pág. 47). O desenvolvimento de aplicações para dispositivos móveis é uma área em constante evolução sendo que, devido às características dos próprios sistemas operativos, leva à existência de linguagens e ambientes de desenvolvimento, diferentes para cada plataforma. Para o desenvolvimento deste módulo foi selecionado a tecnologia *Xamarin*, utilizando a linguagem NET.

O *Xamarin* é uma plataforma de desenvolvimento de aplicações móveis multiplataforma nativas, desenvolvida pela fundação Mono e posteriormente adquirida pela Microsoft em 2015. O *Xamarin*, disponibiliza uma abstração ao desenvolvimento de aplicações móveis, através de uma camada genérica, comum aos vários dispositivos, permitindo, assim, que seja possível reutilizar uma grande parte do código entre plataformas. A Figura 5-12 representa essa intenção de reutilização ao permitir a partilha de uma grande quantidade de código, normalmente relativo à função da aplicação em si, permitindo que a parte específica da plataforma seja organizado em pequenos blocos específicos, reduzindo assim o tempo e o esforço no desenvolvimento aplicacional.

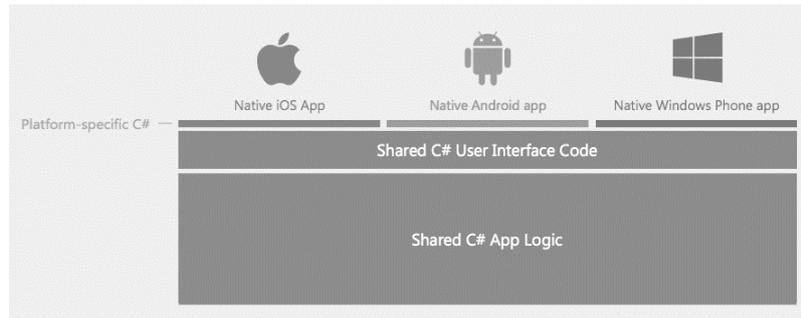


FIGURA 5-12 – PLATAFORMA XAMARIM

Na plataforma Android, o código C# é compilado para uma linguagem intermédia (CIL), que é executada pelo ambiente de execução MONO, o que leva que a aplicação não seja executada de forma nativa, mas ocorre de forma equivalente sendo, no entanto, dependente da performance e de restrições do ambiente virtual MONO.

Para a utilização da técnica *Random Forest*, foi utilizada a biblioteca *RandomForest.Lib* (disponível em <https://github.com/pavlov/RandomForest>), de forma a otimizar e reduzir a dificuldade de desenvolvimento, uma vez que o âmbito desta tese não é a implementação matemática da *Random Forest* per se, mas sim a inferência de contexto e sua distribuição. Outra das razões foi para evitar situações de erro, causadas por erros de implementação incorreta dos algoritmos e não por treino incorreto.

Exemplo de utilização da biblioteca RandomForest.Lib

```
// Parametros de construção da "Floresta"
ForestGrowParameters p = new ForestGrowParameters
{
    ExportDirectoryPath //local para armazenar o resultado de treino
    ResolutionFeatureName //Nome da característica para treinar],
    ItemSubsetCountRatio //rácio de elementos para separar em Treino/Validação automaticamente
    TrainigDataPath //localização do ficheiro com os dados para treino
    MaxItemCountInCategory //número máximo de elementos por categoria (recomendado 5-10)
    TreeCount //número de árvores a gerar
    SplitMode //método de divisão de árvore
};

// Método para treinar a Árvore
IForest forest = ForestFactory.Create();
forest.GrowAsync(p);
```

Para poder realizar o treino da árvore é necessária a aquisição dos dados, nomeadamente, realizar o mapeamento das divisões em que se pretende realizar a localização interna. Assim, foi

desenvolvida uma aplicação com o único objetivo de registrar os RSS (*Received Signal Strength/* força de sinal recebida) da rede WiFi, e recolher várias amostras, associando-as a uma determinada divisão. Para a realização deste *survey* foram utilizadas as capacidades nativas do Android, nomeadamente, a capacidade de receber a informação das redes disponíveis:

```
receiverWifi = new WifiReceiver();
registerReceiver(receiverWifi, new IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));
```

Em seguida, a aplicação irá estar pronta a receber, periodicamente, a lista das redes WiFi disponíveis com a informação necessária associada sendo que, o processo inicia-se com a seleção da sala a que pretende associar as amostras. Em seguida, o sistema iniciará a recolha de amostras para uso posterior.

Exemplo de aquisição de dados WiFi

```
wifiList = mainWifi.getScanResults(); //obter lista de redes disponíveis
String sampleID = UUID.randomUUID().toString();

for(int i = 0; i < wifiList.size(); i++)
{
    ScanResult wifi = wifiList.get(i); //obter os detalhes de cada uma
    WifiInfo info = new WifiInfo();

    //registar a informação de cada uma das informações
    info.setBssid(wifi.BSSID);
    info.setSignal(wifi.level);
    info.setSsid(wifi.SSID);

    //guardar para posterior uso
    ListWifiSamples.Add(selectedRoom, sampleID ,info);
}
```

Associado a esta recolha, será registado, também, um identificador único de cada recolha, o que permite saber quais os sinais disponíveis em cada uma das posições. Assim, com base nos dados recolhidos, é possível iniciar o treino da floresta, utilizando o método já descrito anteriormente. Após a realização do treino, é possível realizar testes de categorização.

Exemplo de identificação da sala

```
var item = forest.CreateItem();
item.SetValue("X", "00:1c:f0:07:92:9c");
item.SetValue("Y", -82.4362);
String Sala = forest.Resolve(item);
```

Uma das vantagens da utilização desta biblioteca é a capacidade de treino em dispositivos diferentes dos que irão utilizar o modelo para inferência. Assim, é possível treinar um modelo, com um conjunto de salas bastante alargado, num servidor com maiores capacidades computacionais e, posteriormente, carregar os ficheiros de treino, previamente obtidos, e começar a realizar a localização corretamente.

Relativamente ao desenvolvimento da Fonte Contextual, tal como referido no capítulo anterior, foi utilizada uma arquitetura de micro serviços, pelo que ,no caso desta fonte contextual foi implementada como um serviço sem interface, que será iniciado sempre que receber a notificação *ADAPT.InternalLocationContext.START*, e os eventos serão publicados sob a seguinte evento *Locations.ACTION_ADAPT_LOCATIONS*, que será invocado sempre que ocorrer uma alteração na localização interna, e ao evento *ADAPT.InternalLocationContext.GET_LOCATION*, que retorna, uma única vez, a localização atual do utilizador.

TABELA 5-2 – TIPO DE EVENTOS CONSUMIDOS E PUBLICADOS PELA FONTE CONTEXTUAL LOCALIZAÇÃO INTERNA

ADAPT.INTERNALLOCATIONCONTEXT.START	EVENTO QUE INDICA A NECESSIDADE DE INÍCIO DE PUBLICAÇÃO DE CONTEXTO DE LOCALIZAÇÃO INTERNA DE FORMA CONTINUA PUBLICADOS PERIODICAMENTE SEMPRE QUE OCORREREM ALTERAÇÕES
ADAPT.INTERNALLOCATIONCONTEXT.GET_LOCATION	EVENTO QUE INDICA A NECESSIDADE DE IDENTIFICAR A LOCALIZAÇÃO INTERNA DO UTILIZADOR E PUBLICAR A MESMA UMA ÚNICA VEZ NO CANAL DE COMUNICAÇÃO.
LOCATIONS.ACTION_ADAPT_LOCATIONS	EVENTO ONDE SERÃO PUBLICADOS OS EVENTOS DE LOCALIZAÇÃO INTERNA.

No caso do evento, *Locations.ACTION_ADAPT_LOCATIONS*, ao evento será associado um conjunto de informação que irá permitir identificar a localização do utilizador. Essa estrutura é apresentada na Tabela 5-3.

TABELA 5-3 – ESPECIFICAÇÃO DA INFORMAÇÃO DE LOCALIZAÇÃO INTERNA

CAMPO	TIPO	DESCRIÇÃO
_ID	INTEGER	ID DO EVENTO, AUTO INCREMENTÁVEL
TIMESTAMP	REAL	DATA DE CRIAÇÃO DA LOCALIZAÇÃO
DEVICE_ID	TEXT	ID DA FONTE CONTEXTUAL
LABEL	TEXT	NOME DO LOCAL (DIVISÃO, ETC) ONDE O UTILIZADOR SE ENCONTRA
LATITUDE	REAL	LATITUDE DO UTILIZADOR SE ASSOCIADA COM O LOCAL
LONGITUDE	REAL	LONGITUDE DO UTILIZADOR SE ASSOCIADA COM O LOCAL
ALTITUDE	REAL	ALTITUDE DO UTILIZADOR SE ASSOCIADA COM O LOCAL
ACCURACY	INTEGER	PRECISÃO ESTIMADA RELATIVO A INFERÊNCIA DO CONTEXTO
PROVIDER	TEXT	RANDOMFOREST/GPS/ETC

Reconhecimento de atividade humana

O desafio do reconhecimento da atividade humana foi já apresentado anteriormente, nomeadamente na secção 2.5.2, (pág. 52), onde foi apresentada uma abordagem que implementa uma rede neuronal convolucional, para o reconhecimento da atividade do utilizador, a partir dos dados do acelerómetro. Devido à complexidade de implementação que a abordagem selecionada apresenta, à semelhança da situação anterior, optou-se por utilizar uma biblioteca de computação, que já tenha implementadas as funções que se pretende, de forma a evitar erros de implementação.

Baseando-se nessa decisão, a opção escolhida foi o *TensorFlow*, uma biblioteca *open-source* da Google dedicada à computação numérica utilizando grafos. De forma semelhante à situação anterior, também para este caso é necessário a recolha de dados para poder realizar o treino do modelo. Devido à complexidade de obtenção de um conjunto de treino válido, diversificado e categorizado, utilizou-se o *dataset* de treino *Actitraker* (Lockhart et al., 2011), que consiste num conjunto de 6 atividades diárias (correr, andar, subir escadas, descer escadas,

sentado e em pé), recolhidas em ambiente controlado, por 36 pessoas com um smartphone no bolso, com uma taxa de amostragem de 20Hz (20 amostras por segundo). Este *dataset* apresenta a distribuição ilustrada na Figura 5-13. É possível verificar que o maior número de amostras recolhidas diz respeito a andar e correr, o que era esperado pois é a atividade mais comum do ser humano.

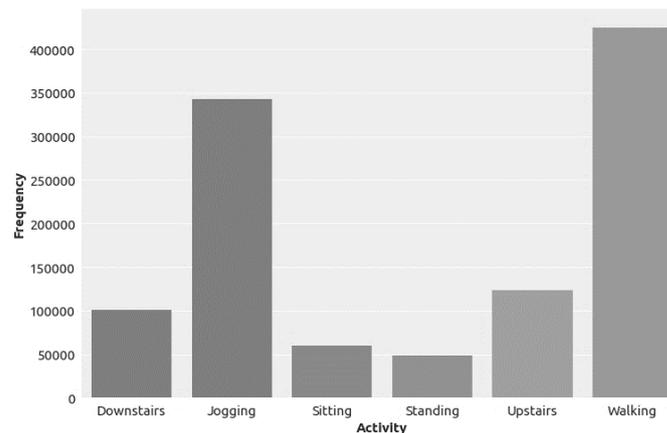


FIGURA 5-13 – DISTRIBUIÇÃO DOS DADOS DA AMOSTRA EM RESPEITO À ATIVIDADE (LOCKHART ET AL., 2011)

Em seguida, utilizando as bibliotecas, *TensorFlow*, *Numpy*, *Matplotlib*, e *Pandas* implementaram-se os códigos necessários para a realização do treino da CNN. No entanto, antes de iniciar o processo de treino, é benéfico realizar algumas operações prévias de visualização e pré processamento.

Implementação de funções de leitura, normalização e visualização dos dados

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import tensorflow as tf

def read(file_path):
    column_names = ['user-id','activity','timestamp', 'x-axis', 'y-axis', 'z-axis']
    data = pd.read_csv(file_path,header = None, names = column_names)
    return data

def feature_normalization(dataset):
    mu = np.mean(dataset,axis = 0)
    sigma = np.std(dataset,axis = 0)
    return (dataset - mu)/sigma

def plot_axis(ax, x, y, title):
    ax.plot(x, y)
    ax.set_title(title)
```

```

ax.xaxis.set_visible(False)
ax.set_ylim([min(y) - np.std(y), max(y) + np.std(y)])
ax.set_xlim([min(x), max(x)])
ax.grid(True)
def plot_activity(activity,data):
    fig, (ax0, ax1, ax2) = plt.subplots(nrows = 3, figsize = (15, 10), sharex = True)
    plot_axis(ax0, data['timestamp'], data['x-axis'], 'x-axis')
    plot_axis(ax1, data['timestamp'], data['y-axis'], 'y-axis')
    plot_axis(ax2, data['timestamp'], data['z-axis'], 'z-axis')
    plt.subplots_adjust(hspace=0.2)
    fig.suptitle(activity)
    plt.subplots_adjust(top=0.90)
    plt.show()

```

Assim, com base nestas funções, é possível realizar a leitura e normalização dos ficheiros utilizando a função *read*, para ler e a função *feature_normalization* para a normalização dos dados.

Exemplo de leitura e normalização dos dados

```

dataset = read ('actitracker_raw.txt')
dataset['x-axis'] = feature_normalization(dataset['x-axis'])
dataset['y-axis'] = feature_normalization(dataset['y-axis'])
dataset['z-axis'] = feature_normalization(dataset['z-axis'])

```

Em seguida podemos visualizar cada componente do acelerómetro para as diferentes atividades. Por exemplo, com o seguinte código é possível visualizar 10 segundos (200 amostras 10 x 20hz) de sinal para cada atividade, onde é possível verificar já algumas variações no sinal, consoante a atividade.

Visualização dos dados em forma gráfica

```

for activity in np.unique(dataset["activity"]):
    subset = dataset[dataset["activity"] == activity][:200]
    plot_activity(activity,subset)

```

Com base neste código é possível obter as seguintes análises da Figura 5-14:

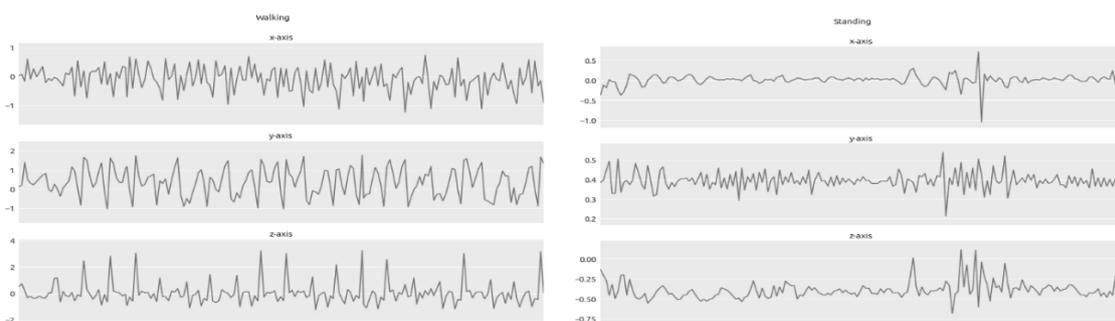


FIGURA 5-14 – VARIAÇÃO DO SINAL NAS DIFERENTES ATIVIDADES CAMINHAR (À ESQUERDA) PARADO (À DIREITA)

Após o processo de normalização, devem preparar-se os dados para o formato compatível com o modelo da CNN, sendo necessário implementar algumas funções que permitem criar segmentos fixos a partir do sinal contínuo.

Função que implementa a seleção de um conjunto de amostras no tempo

```
def windows(data, size):
    start = 0
    while start < data.count():
        yield start, start + size
        start += (size / 2)

def segment_signal(data, window_size = 90):
    segments = np.empty((0, window_size, 3))
    labels = np.empty((0))
    for (start, end) in windows(data["timestamp"], window_size):
        x = data["x-axis"][start:end]
        y = data["y-axis"][start:end]
        z = data["z-axis"][start:end]
        if(len(dataset["timestamp"][start:end]) == window_size):
            segments = np.vstack([segments, np.dstack([x, y, z])])
            labels = np.append(labels, stats.mode(data["activity"][start:end])[0][0])
    return segments, labels
```

Esta função irá recolher amostras com o tamanho de 90 pontos, que corresponde a uma amostra temporal de 4,5 segundos de dados e, como o movimento da janela na amostra é de 45 pontos, efetivamente estamos a avançar a cada 2,25 segundos no sinal; para seleção da janela a classificação será obtida pela mais frequente nessa seleção.

Exemplo de implementação das funções

```
segments, labels = segment_signal(dataset)
labels = np.asarray(pd.get_dummies(labels), dtype = np.int8)
reshaped_segments = segments.reshape(len(segments), 1, 90, 3)

train_test_split = np.random.rand(len(reshaped_segments)) < 0.70
train_x = reshaped_segments[train_test_split]
train_y = labels[train_test_split]
test_x = reshaped_segments[~train_test_split]
test_y = labels[~train_test_split]
```

Após colocar os dados num formato que seja compatível com a rede neuronal, é necessário criar os conjuntos de treino e avaliação, de forma aleatória, numa distribuição de 70/30. Para a implementação da CNN, propriamente dita, implementou-se um conjunto de funções para inicializar as variáveis do *TensorFlow*.

Implementação do TensorFlow para entrada e saída de dados

```

X = tf.placeholder(tf.float32, shape=[None,input_height,input_width,num_channels])
Y = tf.placeholder(tf.float32, shape=[None,num_labels])

c = apply_depthwise_conv(X, kernel_size, num_channels, depth)
p = apply_max_pool(c, 20, 2)
c = apply_depthwise_conv(p, 6, depth*num_channels, depth//10)

shape = c.get_shape().as_list()
c_flat = tf.reshape(c, [-1, shape[1] * shape[2] * shape[3]])

f_weights_l1 = weight_variable([shape[1] * shape[2] * depth * num_channels * (depth//10),
num_hidden])
f_biases_l1 = bias_variable([num_hidden])
f = tf.nn.tanh(tf.add(tf.matmul(c_flat, f_weights_l1), f_biases_l1))

out_weights = weight_variable([num_hidden, num_labels])
out_biases = bias_variable([num_labels])
y_ = tf.nn.softmax(tf.matmul(f, out_weights) + out_biases)

loss = -tf.reduce_sum(Y * tf.log(y_))
optimizer = tf.train.GradientDescentOptimizer(learning_rate = learning_rate).minimize(loss)

correct_prediction = tf.equal(tf.argmax(y_,1), tf.argmax(Y,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

```

Para podermos reutilizar o modelo treinado noutro dispositivo, é necessário exportá-lo.

Exportação do Modelo TensorFlow

```

from tensorflow.python.tools import freeze_graph
from tensorflow.python.tools import optimize_for_inference_lib

freeze_graph.freeze_graph(input_graph = "../har.pbtxt", input_saver = "",
input_binary = False, input_checkpoint = "../har.ckpt", output_node_names = "y_",
restore_op_name = "save/restore_all", filename_tensor_name = "save/Const:0",
output_graph = "frozen_har.pb", clear_devices = True, initializer_nodes = "")

input_graph_def = tf.GraphDef()
with tf.gfile.Open(output_frozen_graph_name, "r") as f:
    data = f.read()
    input_graph_def.ParseFromString(data)

output_graph_def = optimize_for_inference_lib.optimize_for_inference(
    input_graph_def,
    ["input"],
    ["y_"],
    tf.float32.as_datatype_enum)

f = tf.gfile.FastGFile("optimized_frozen_har.pb", "w")
f.write(output_graph_def.SerializeToString())

```

Após cumpridos todos os requisitos para implementar o treino resta, apenas, implementar o treino em si, utilizando uma configuração de 5 épocas.

Exemplo de treino da CNN com TensorFlow

```
with tf.Session() as session:
    tf.initialize_all_variables().run()
    for epoch in range(training_epochs):
        cost_history = np.empty(shape=[1],dtype=float)
        for b in range(total_batches):
            offset = (b * batch_size) % (train_y.shape[0] - batch_size)
            batch_x = train_x[offset:(offset + batch_size), :, :]
            batch_y = train_y[offset:(offset + batch_size), :]
            _, c = session.run([optimizer, loss],feed_dict={X: batch_x, Y : batch_y})
            cost_history = np.append(cost_history,c)
        print "Epoch: ",epoch," Training Loss: ",np.mean(cost_history)," Training Accuracy: ",
            session.run(accuracy, feed_dict={X: train_x, Y: train_y})

    tf.train.write_graph(session.graph_def, '.', './har.pbtxt')
    saver.save(session,save_path = "./har.ckpt")

print "Testing Accuracy:", session.run(accuracy, feed_dict={X: test_x, Y: test_y})
```

Para a implementação em android, é necessário obter as bibliotecas necessárias para invocar o modelo do *TensorFlow* e, para isso, é necessário importá-lo para o projeto. O primeiro passo é obter a biblioteca correspondente para Android (disponível em <https://ci.tensorflow.org/view/Nightly/job/nightly-android/>), e copiar o ficheiro *libandroid_tensorflow_inference_java.jar*, e as pastas *arm64-v8a*, *armeabi-v7a*, *x86* e *x86_64*, para a pasta *lib* do projeto Android. Por fim, copiar o modelo exportado para a pasta *assets* e adicionar a referência à pasta na variável *jniLibs.srcDirs* no ficheiro *build.gradle*. De forma a realizar a interface com o código nativo, é necessária uma classe que exponha a interface nativa do modelo TensorFlow de forma compatível com o resto da aplicação. De referir que, à data desta tese foi verificada compatibilidade com a compilação 90 desta biblioteca.

Exemplo de interface de acesso à biblioteca nativa TensorFlow

```
import android.content.Context;
import android.content.res.AssetManager;
import org.tensorflow.contrib.android.TensorFlowInferenceInterface;

public class ActivityInference {
    static {
        System.loadLibrary("tensorflow_inference");
    }
}
```

```

private static ActivityInference activityInferenceInstance;
private TensorFlowInferenceInterface inferenceInterface;
private static final String MODEL_FILE = "file:///android_asset/optimized_har.pb";
private static final String INPUT_NODE = "input";
private static final String[] OUTPUT_NODES = {"y_"};
private static final String OUTPUT_NODE = "y_";
private static final long[] INPUT_SIZE = {1,270};
private static final int OUTPUT_SIZE = 6;
private static AssetManager assetManager;

public static ActivityInference getInstance(final Context context)
{
    if (activityInferenceInstance == null)
    {
        activityInferenceInstance = new ActivityInference(context);
    }
    return activityInferenceInstance;
}

public ActivityInference(final Context context) {
    this.assetManager = context.getAssets();
    inferenceInterface = new TensorFlowInferenceInterface(assetManager, MODEL_FILE);
}

public float[] getActivityProb(float[] input_signal)
{
    float[] result = new float[OUTPUT_SIZE];
    inferenceInterface.feed(INPUT_NODE, input_signal, INPUT_SIZE);
    inferenceInterface.run(OUTPUT_NODES);
    inferenceInterface.fetch(OUTPUT_NODE, result);
    return result;
}
}

```

E, com isto, é possível realizar a predição da atividade, invocando o modelo do *TensorFlow*, com a informação atual do acelerómetro.

Exemplo de inferência usando o TensorFlow numa aplicação Android

```

private final int N_SAMPLES = 90;
                                (...)

if(x.size() == N_SAMPLES && y.size() == N_SAMPLES && z.size() == N_SAMPLES) {
    // copiar todos os valores para um vector N_Samples*3
    input_signal.addAll(x);
    input_signal.addAll(y);
    input_signal.addAll(z);

    // realizar a inferência com o modelo
    float[] results = activityInference.getActivityProb(toFloatArray(input_signal));
}

```

Relativamente ao desenvolvimento desta fonte contextual, à semelhança da proposta anterior, foi implementada baseando-se num microserviço sem interface, que será iniciado sempre que receber a notificação *ADAPT.ActivityRecognitionContext.START*.

As alterações de atividade serão publicadas sob o seguinte evento *ADAPT.ACTIVITY_RECOGNITION_CHANGE*, e ao evento *ADAPT.ActivityRecognitionContext.GET_ACTIVITY*, que retorna uma única vez a atividade atual do utilizador.

TABELA 5-4 – TIPO DE EVENTOS CONSUMIDOS E PUBLICADOS PELA FONTE CONTEXTUAL LOCALIZAÇÃO INTERNA

ADAPT.ACTIVITYRECOGNITIONCONTEXT.START	EVENTO QUE INDICA A NECESSIDADE DE INÍCIO DE PUBLICAÇÃO DE CONTEXTO DE RECONHECIMENTO DE ATIVIDADE DO UTILIZADOR PUBLICADOS PERIODICAMENTE
ADAPT.ACTIVITYRECOGNITIONCONTEXT.GET_ACTIVITY	EVENTO QUE INDICA A NECESSIDADE DE IDENTIFICAR A ATIVIDADE DO UTILIZADOR E PUBLICAR A MESMA UMA ÚNICA VEZ NO CANAL DE COMUNICAÇÃO.
ADAPT.ACTIVITY_RECOGNITION_CHANGE	EVENTO ONDE SERÃO PUBLICADOS OS EVENTOS DE ATIVIDADE DO UTILIZADOR.

No caso do evento *ADAPT.ACTIVITY_RECOGNITION_CHANGE*, esta será associado um conjunto de informação que irá permitir identificar a localização do utilizador. E essa estrutura é apresentada na Tabela 5-3.

TABELA 5-5 – ESPECIFICAÇÃO DA INFORMAÇÃO DE LOCALIZAÇÃO INTERNA

CAMPO	TIPO	DESCRIÇÃO
_ID	INTEGER	ID DO EVENTO, AUTO INCREMENTÁVEL
TIMESTAMP	REAL	DATA DE CRIAÇÃO DA LOCALIZAÇÃO
DEVICE_ID	TEXT	ID DA FONTE CONTEXTUAL
ACTIVITY_NAME	TEXT	ATIVIDADE INFERIDA EM MODO DE DESCRIÇÃO HUMANA, (EX. "WALKING", "RUNNING", ETC)
ACTIVITY_TYPE	INTEGER	TIPO DE ATIVIDADE
CONFIDENCE	INTEGER	PERCENTAGEM DE CONFIANÇA DA PREDIÇÃO (0-100)
ACTIVITIES	TEXT	LISTA DE ATIVIDADES POSSÍVEIS ALTERNATIVAS

5.4. Considerações finais do capítulo

Este capítulo pretende apresentar a arquitetura desenvolvida para validar o modelo contextual antecipatório, detalhando e fundamentando as decisões tomadas. De forma a aumentar a compreensão sobre o modelo ADAPT, este capítulo iniciou-se com a apresentação da arquitetura, partindo de uma visão abstrata e, prosseguindo com a especificação de toda a arquitetura, à medida que vamos avançando no capítulo. Concluiu-se com a apresentação da visão lógica e da implementação de cada um dos módulos principais, com exemplos de implementação em código para cada um dos conceitos mais importantes.

O sistema ADAPT, tem como principal característica dar suporte a um conjunto de fontes contextuais diversas e heterogêneas utilizando, para o efeito, um canal de comunicação comum a todos os intervenientes, que este pode ser implementado através de um *broker*. Este canal de comunicação é, de uma forma genérica, a *cola* que liga todas as peças do sistema, permitindo que a gestão e a comunicação entre aplicações – ADAPT – fontes contextuais, funcione independentemente da implementação particular de cada uma delas.

No capítulo seguinte, será apresentado o caso de estudo e avaliação, realizado para comprovar o funcionamento da arquitetura proposta e, assim, validar o modelo conceptual antecipatório defendido nesta tese.

6. CASO DE ESTUDO E AVALIAÇÃO

Neste capítulo é apresentada a avaliação e validação do ADAPT e da arquitetura de suporte desenvolvida, suportada pela apresentação do caso de estudo que serviu de base à realização de experiências reais. A avaliação da arquitetura é apresentada sob uma vista qualitativa ao nível da interação e quantitativas ao nível do desempenho e precisão do sistema.

6.1. Enquadramento

No sentido de avaliar a arquitetura proposta e, assim, comprovar o modelo definido nesta tese, foi desenvolvida uma aplicação móvel que implementa o ADAPT proporcionando ao utilizador, uma experiência de utilização mais eficiente. Esta validação consiste no desenvolvimento de uma aplicação para auxiliar os alunos da UTAD no seu dia-a-dia, apresentando-lhes informação que é contextualmente relevante e, em algumas situações, assumindo um comportamento antecipatório.

Este capítulo inicia-se com apresentação do domínio do problema a resolver e em que situações o contexto é benéfico para a aplicação sendo, posteriormente, apresentado um conjunto de requisitos funcionais e contextuais, necessários para a implementação da aplicação móvel. O capítulo prossegue com a apresentação dos fluxos de interação da aplicação nas diversas funcionalidades e de que forma estas são alteradas com a implementação do ADAPT.

Após esta revisão do domínio do problema, é apresentado, em maior detalhe, o processo de desenvolvimento e implementação, começando por uma visão técnica do processo e terminando com a apresentação da avaliação e validação do modelo. Neste ponto, são apresentados os resultados obtidos, de que forma os requisitos da aplicação foram cumpridos e, principalmente, se a arquitetura proposta para o modelo contextual antecipatório permite que as aplicações beneficiem de informação contextual.

6.1.1. Domínio do problema

O SIDE é um sistema de informação com suporte web, existente Universidade de Trás-os-Montes e Alto Douro que entre outras funcionalidades, visa disponibilizar o acesso à informação académica e curricular dos alunos (Barbosa, Alves, & Barroso, 2011). Apesar de possuir outros módulos, no âmbito desta tese, consideraremos apenas o portal web – versão alunos. Esta aplicação destina-se a permitir a interação entre docentes e alunos, através da partilha de informação relativa às atividades letivas, como as apresentadas na Tabela 6-1.

TABELA 6-1 – RESUMO DE SERVIÇOS PRESTADOS PELO SIDE PARA OS ALUNOS (ADAPTADO DE BARBOSA ET AL., 2011)

ANO LECTIVO	DEFINIÇÃO DO PLANO CURRICULAR DO CURSO, CARGA LETIVA, E TIPOLOGIAS DE AULAS
HORÁRIO	DEFINIÇÃO DOS HORÁRIOS SEMANAIS, INSCRIÇÕES EM TURMAS
EXAMES	DATAS DE AVALIAÇÃO, INSCRIÇÕES E RESULTADOS FINAIS
TURMAS	INSCRIÇÕES EM TURMAS E DISCIPLINAS
AVISOS	AVISOS E NOTÍCIAS DA PLATAFORMA, UNIDADES CURRICULARES
ASSIDUIDADE	VISUALIZAÇÃO DA ASSIDUIDADE DO ALUNO, COM POSSIBILIDADE DE JUSTIFICAÇÃO DE FALTAS
TRABALHOS	SUBMISSÃO E CONSULTA DE RESULTADOS DOS TRABALHOS RELATIVOS A UMA UNIDADE CURRICULAR
AVALIAÇÕES	INSCRIÇÃO E RESULTADOS DA AVALIAÇÕES PERIÓDICAS DE UMA UNIDADE CURRICULAR

Devido à importância que este serviço representa para os alunos, uma vez que este é o ponto central de toda a informação académica, é importante que o acesso a esta informação seja realizado da forma mais eficaz e, principalmente, através do meio mais prático para o aluno.

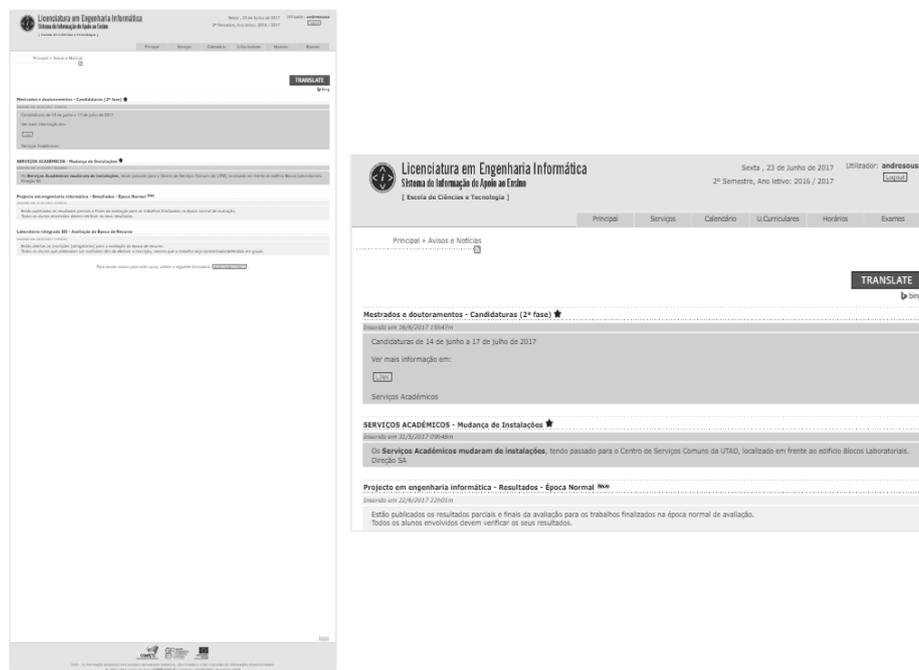


FIGURA 6-1 – ASPETO DO PORTAL SIDE UTILIZANDO UM SMARTPHONE

Como é possível verificar, a utilização deste portal num dispositivo móvel, é um pouco complicada, principalmente devido à constante necessidade de realizar zoom na página para

selecionar o menu pretendido, algo que também se torna mais complexo devido a esta ser otimizada para a utilização com um rato. Assim, o tempo necessário para obter, do SIDE, uma determinada informação, pode alongar-se, sobretudo devido à complexidade e quantidade de informação do SIDE, incluindo menus dentro de menus, com hierarquias significativamente complexas, e aos erros provocados pela interação com o SIDE num smartphone. Este alongar do tempo e a necessidade de repetição devido a erros, leva a que os alunos sintam que não conseguem encontrar a informação que necessitam e, por esse motivo, fiquem frustrados.

Através da utilização da informação contextual é possível fazer com que a interação, dos alunos com a aplicação, seja muito mais eficiente, uma vez que é possível reduzir o tempo que o utilizador perde na procura da informação que necessita. Assim, propomo-nos a construir uma aplicação móvel que permita ao aluno consultar a informação de forma mais simples e rápida. No entanto, no âmbito desta tese, apenas foram selecionados algumas das funcionalidades do SIDE para implementar na versão móvel, visto que se trata de uma versão de prova de conceito.

Com esse objetivo, propomos a implementação das seguintes situações, que foram identificadas como sendo as que mais beneficiam de adaptação contextual:

- **Atingir limite de faltas** – em diversas situações e, por diversos motivos, os alunos atingem o limite de faltas em várias disciplinas sendo, por isso, a procura da informação relativa à sua situação atual, uma das atividades mais realizadas por eles. Deste modo, a aplicação pode ter um comportamento proactivo e, quando deteta que o aluno está a atingir o limite de faltas, poderá notificá-lo e propor-lhe a ativação de um alarme para a aula, caso esta seja a primeira aula do dia.
- **Inscrição/Submissão de trabalhos ou frequências** – em muitas disciplinas, a inscrição de trabalhos ou frequências têm um prazo limite. Assim, a aplicação pode informar o aluno do tempo restante para uma determinada submissão notificando-o, quando se aproximar o final do prazo e o aluno ainda não o tiver feito.

- **Informação sobre a disciplina** – em vários momentos é necessário procurar informação na página da disciplina, seja na lista de downloads, contactos dos docentes ou, mesmo, nas datas de frequências e exames. A aplicação pode apresentar um comportamento antecipatório e, consoante o aluno se encontra na aula, ou a dirigir-se para o local, apresentar-lhe informação relevante sobre ela.

6.1.2. Requisitos aplicacionais

De forma a permitir a manutenção de um nível de funcionalidade mínimo, mesmo em situações que não seja possível otimizar a interface com base no contexto, é necessário que a aplicação atenda a alguns requisitos funcionais não contextuais (**F**), que devem ser cumpridos, independentemente da utilização ou não de informação contextual para a interface. Deste modo, é possível definir uma linha base de funcionamento, que é passível de ser melhorada e, assim, perceber se as alterações à interação propostas pelas otimizações contextuais estão a melhorar ou a prejudicar a experiência de uso. A lista desses requisitos encontra-se descrita na Tabela 6-2.

TABELA 6-2 – LISTA DE REQUISITOS FUNCIONAIS NÃO CONTEXTUAIS

F1	O UTILIZADOR DEVE SER CAPAZ DE EFETUAR LOGIN NA APLICAÇÃO PARA ACEDER AOS SEUS DADOS ACADÉMICOS;
F2	O UTILIZADOR DEVE PODER CONSULTAR AS INFORMAÇÕES DA DISCIPLINA;
F3	O UTILIZADOR DEVE PODER OBTER CONSEGUIR NAVEGAR NA APLICAÇÃO PARA OBTER A LISTA DE DISCIPLINAS;
F4	O UTILIZADOR DEVE PODER CONSULTAR A SUA SITUAÇÃO DE ASSIDUIDADE ÀS AULAS;
F5	O UTILIZADOR DEVE PODER CONSULTAR AS DATAS DE ENTREGA DE TRABALHOS E FREQUÊNCIAS;
F6	O UTILIZADOR DEVE PODER CONSULTAR O HORÁRIO DIÁRIO E SEMANAL PESSOAL;
F7	A APLICAÇÃO DEVE FUNCIONAR MESMO QUE NÃO POSSUA INFORMAÇÃO CONTEXTUAL;
F8	A APLICAÇÃO DEVE POSSUIR UMA VERSÃO NÃO CONTEXTUAL PARA TODAS AS INTERFACES DINÂMICAS;
F9	A APLICAÇÃO DEVE PODER FUNCIONAR DE FORMA INDEPENDENTE À LIGAÇÃO À INTERNET

No mesmo sentido, existe um conjunto de requisitos funcionais contextuais (**C**) que também deve ser atendido por parte da aplicação desenvolvida, de forma a acrescentar uma otimização ao tipo de utilização e interação, típico da aplicação. Uma lista destes requisitos encontra-se na Tabela 6-3. Alguns destes requisitos têm como objetivo a apresentação de uma visão computacional antecipatória, com vista a disponibilizar a informação que o utilizador necessita, quando este realmente a precisa.

TABELA 6-3 – LISTA DE REQUISITOS FUNCIONAIS CONTEXTUAIS

C1	A APLICAÇÃO DEVE INFERIR CORRETAMENTE A LOCALIZAÇÃO DO UTILIZADOR
C2	A APLICAÇÃO DEVE AGIR PROACTIVAMENTE PARA NOTIFICAR O UTILIZADOR ACERCA DA SUA ASSIDUIDADE
C3	A APLICAÇÃO DEVE NOTIFICAR O UTILIZADOR SOBRE AS AULAS DO DIA SEGUINTE SE ESTE NÃO SE ENCONTRAR EM CASA APÓS A MEIA NOITE
C4	A APLICAÇÃO DEVE AGIR PROACTIVAMENTE PARA NOTIFICAR O UTILIZADOR SOBRE A SUBMISSÃO DE TRABALHOS
C5	A APLICAÇÃO DEVE AGIR PROACTIVAMENTE PARA NOTIFICAR O UTILIZADOR SOBRE A PROXIMIDADE DAS FREQUÊNCIAS E AVALIAÇÕES
C6	A APLICAÇÃO DEVE INFERIR A PRESENÇA DO UTILIZADOR NA SALA DE AULA A PARTIR DO SEU CONTEXTO
C7	A APLICAÇÃO DEVE APRESENTAR DE FORMA INSTANTÂNEA A INFORMAÇÃO RELATIVA ÀS PRÓXIMAS DISCIPLINAS QUANDO NECESSÁRIAS
C8	A APLICAÇÃO DEVE APRESENTAR ACESSO RÁPIDO À INFORMAÇÃO DA DISCIPLINA CONSOANTE AS NECESSIDADES CONTEXTUAIS NESSE INSTANTE.

A informação contextual necessária para permitir dar resposta a estes conjuntos de requisitos, consegue-se restringir aos seguintes tipos de contexto: localização interna no edifício; deteção de tipo de atividade e movimento; informação académica curricular. Com base nesta informação contextual, é possível à aplicação inferir se o aluno está ou não na aula, num determinado momento, pela deteção da localização na sala e ausência de atividade física durante um determinado período de tempo.

6.1.3. Interação com a aplicação

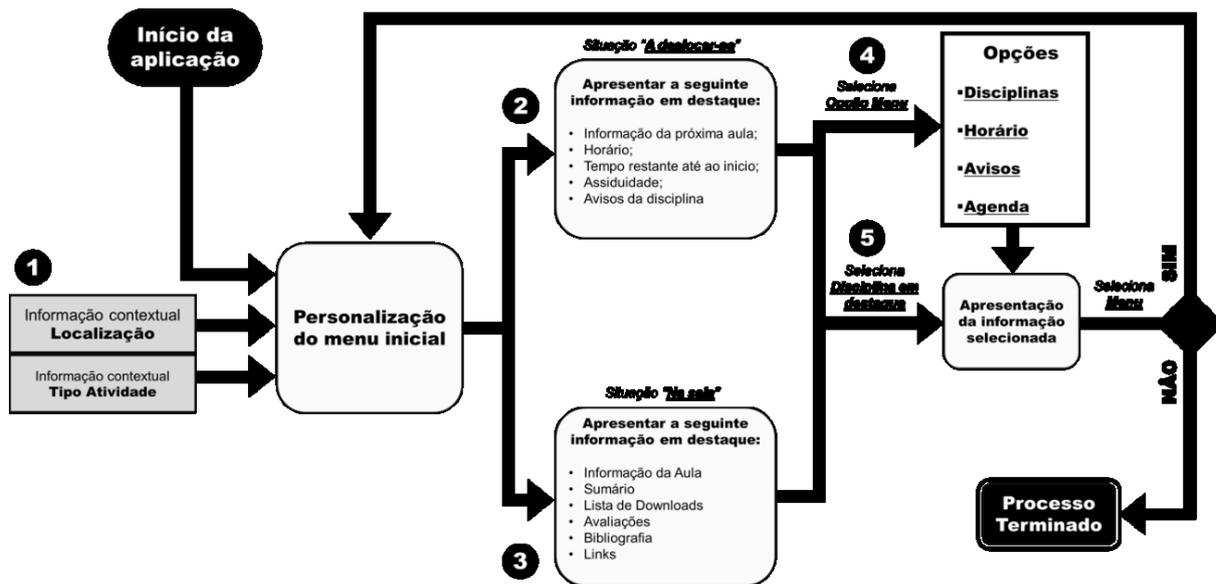
A aplicação desenvolvida para a validação do ADAPT possui um conjunto de interações que podem beneficiar do uso de informação contextual. De forma a compreender o impacto das alterações contextuais, é necessário compreender corretamente o fluxo base do processo. Na primeira inicialização da aplicação, é pedido ao utilizador para preencher os dados de login para a aplicação poder aceder à informação do utilizador, nos serviços de informação académica, e criar uma cópia local para utilização posterior. Este é um processo trivial, pelo que não será especificado em maior detalhe, não sendo apresentado nas interações seguintes.



FIGURA 6-2 – REPRESENTAÇÃO GRÁFICA DO FLUXO DE INTERAÇÃO EM MODO NORMAL

O processo normal de utilização da aplicação inicia-se com a apresentação de uma interface com as capacidades da aplicação, podendo o utilizador seleccioná-la para aceder à informação com maior detalhe. De referir que este processo requer interação manual sem ser realizada nenhuma adaptação dinâmica com base no contexto.

Com a introdução do ADAPT no processo de desenvolvimento é possível aceder a um conjunto de informação extra sobre o contexto do utilizador, nomeadamente, ao nível da sua localização, e do tipo de atividade que está a realizar. Com base nesta informação, é possível personalizar a aplicação para apresentar um conjunto de informação variável, consoante as reais necessidades do utilizador, nesse momento.



- 1 SERVIÇOS EXTERNOS DE INFERÊNCIA CONTEXTUAL PERTENCENTES AO ADAPT QUE COMUNICAM A PARTIR DO CANAL DE COMUNICAÇÃO.
- 2 SITUAÇÃO CONTEXTUAL EM QUE O ALUNO NÃO SE ENCONTRA DENTRO DA AULA, PELO QUE AS SUAS NECESSIDADES PODEM SER RELATIVAS À INFORMAÇÃO SOBRE COMO SE DESLOCAR PARA A SALA, OU AVISOS RELATIVOS À DISCIPLINA.
- 3 SITUAÇÃO CONTEXTUAL EM QUE O ALUNO SE ENCONTRA DENTRO DA SALA DE AULA, DESSE MODO AS SUAS NECESSIDADES SÃO REFERENTES A INFORMAÇÃO SOBRE A DISCIPLINA, COMO DOWNLOADS DISPONÍVEIS, BIBLIOGRAFIA, LINKS OU A LISTA DE AVALIAÇÕES.
- 4 OPÇÃO DE APRESENTAR O MENU NORMAL DE NAVEGAÇÃO, CASO O UTILIZADOR NÃO OBTENHA A INFORMAÇÃO PRETENDIDA TEM SEMPRE DISPONÍVEL O MODO HABITUAL DE ACESSO À INFORMAÇÃO.
- 5 CADA UM DOS MENUS PERSONALIZADOS POSSUI UM SUBCONJUNTO DE ACESSO RÁPIDO A INFORMAÇÃO, COMO POR EXEMPLO, ACESSO DIRETO A LISTA DE DOWNLOADS DESSA DISCIPLINA, OU A PAGINA DE AVISOS, OU CONTACTOS DO PROFESSOR.

FIGURA 6-3 – REPRESENTAÇÃO GRÁFICA DO FLUXO DE INTERAÇÃO EM MODO ENRIQUECIDO PELO CONTEXTO

A Figura 6-3, representa o fluxo comum de interação da aplicação, quando existe informação contextual que lhe permita inferir algumas das necessidades do aluno sendo que, para isso, é necessário perceber que ações o utilizador está a realizar, como por exemplo, se se encontra na UTAD, ou a caminho desta, que tipo de atividade está a realizar, está parado ou em movimento, e se já esteve, nesse dia, a frequentar outras disciplinas. Apesar de não representado na figura acima, existe um cenário em que as necessidades do utilizador são um pouco ambíguas, nomeadamente, no fim de uma aula e antes do início de outra, em que o aluno pode usar a aplicação para consultar informação da disciplina onde está ou para procurar informação sobre a próxima. Nessa situação, nomeadamente nos últimos 10 minutos para o fim da disciplina, será apresentada em destaque simples, informação rápida sobre a localização da disciplina seguinte.

No entanto existem outras situações, para além da utilização da aplicação em si, que beneficiam de proatividade sensível ao contexto, nomeadamente, a questão da assiduidade às aulas, e avaliações e submissões de trabalhos. Uma vez que estas têm um impacto direto sobre o aluno, a possibilidade de poder ser proactivo sobre algumas situações e notificar o aluno sobre essa informação de forma direta, pode ter impacto ao nível do aproveitamento do aluno (por exemplo, não ultrapassando o limite de faltas).



FIGURA 6-4 – REPRESENTAÇÃO GRÁFICA DO FLUXO DE INTERAÇÃO DOS EVENTOS DE NOTIFICAÇÃO DE PROXIMIDADE DE UM DETERMINADO EVENTO.

Na Figura 6-4, apresenta-se o fluxo de interação e os eventos automáticos de notificação do aluno sobre um conjunto de situações temporalmente sensíveis, nomeadamente, a aproximação do limite de faltas, ou a submissão de trabalho ou faltas. Estas notificações podem ser realizadas através de alertas ao utilizador ou apenas chamadas de atenção na aplicação em si, que irá variar consoante a gravidade do mesmo.

Este evento automático poderá ainda beneficiar da utilização do contexto localização, se o utilizador ainda se encontrar em casa quando uma disciplina em que este se encontrar prestes a atingir o limite de faltas, o sistema poderá agir de uma forma mais proactiva e antecipatórias

apresentando o aviso de forma sonora para notificar o aluno que está prestes a ultrapassar o limite, e, assim, possivelmente reprovar à disciplina.

Nesta secção, descreveram-se alguns dos fluxos de interação com a aplicação que permitem beneficiar da utilização de informação contextual para melhorar as suas capacidades de interação ou mesmo acrescentar a capacidade proactiva ao próprio sistema. Na secção seguinte, iremos explorar um conjunto de tecnologias de suporte e técnicas de inferência contextual, que nos permitem construir esta informação contextual com maior sucesso.

6.2. Implementação e desenvolvimento

Nas secções anteriores deste capítulo apresentou-se o caso de estudo a realizar, nomeadamente, o domínio da aplicação e os respetivos problemas que este pretende solucionar, e quais os requisitos que esta necessita possuir, e como a utilização de contexto poderá melhorar estas funcionalidades.

Nesta secção, apresenta-se a implementação realizada, numa visão mais orientada ao desenvolvimento da aplicação de caso de uso. Assim, esta secção inicia-se com a apresentação do processo de desenvolvimento utilizado, que permitiu otimizar o processo de desenvolvimento, e com a validação de novas funcionalidades nos equipamentos. Através da automatização dos processos de execução de testes unitários, distribuição, e testes em dispositivos, foi possível reduzir significativamente o tempo perdido.

Posteriormente, são apresentadas as implementações de dois tipos de informação contextual, tendo sido implementados em tecnologias diferentes, de forma a validar o conceito de micro serviços heterógenos que a arquitetura proposta propõe. Por fim, é apresentada a aplicação desenvolvida, com a apresentação das diferentes interfaces e algumas das adaptações implementadas.

6.2.1. Processo de desenvolvimento e testes

Nesta secção será descrito o processo de desenvolvimento, distribuição e testes, utilizado, uma vez que a existência de um processo bem estruturado e automatizado poderá permitir a redução significativa de tempo e deteção de possíveis erros. Para o processo de desenvolvimento da aplicação, foi selecionada uma filosofia de desenvolvimento iterativa entre os processos de desenvolvimento, testes e feedback – a filosofia *agile*.

Esta abordagem iterativa permitiu-nos desenvolver a aplicação e ir testando os conceitos à medida que iam sendo implementados, e assim poder receber o feedback o mais rápido possível, corrigido problemas conforme apareciam. Um dos pontos fundamentais desta abordagem é a automatização, especialmente ao nível de testes de integração e de entrega/instalação da aplicação, dispondo de um processo automatizado, desde o desenvolvimento até a instalação e recolha de feedback e erros.

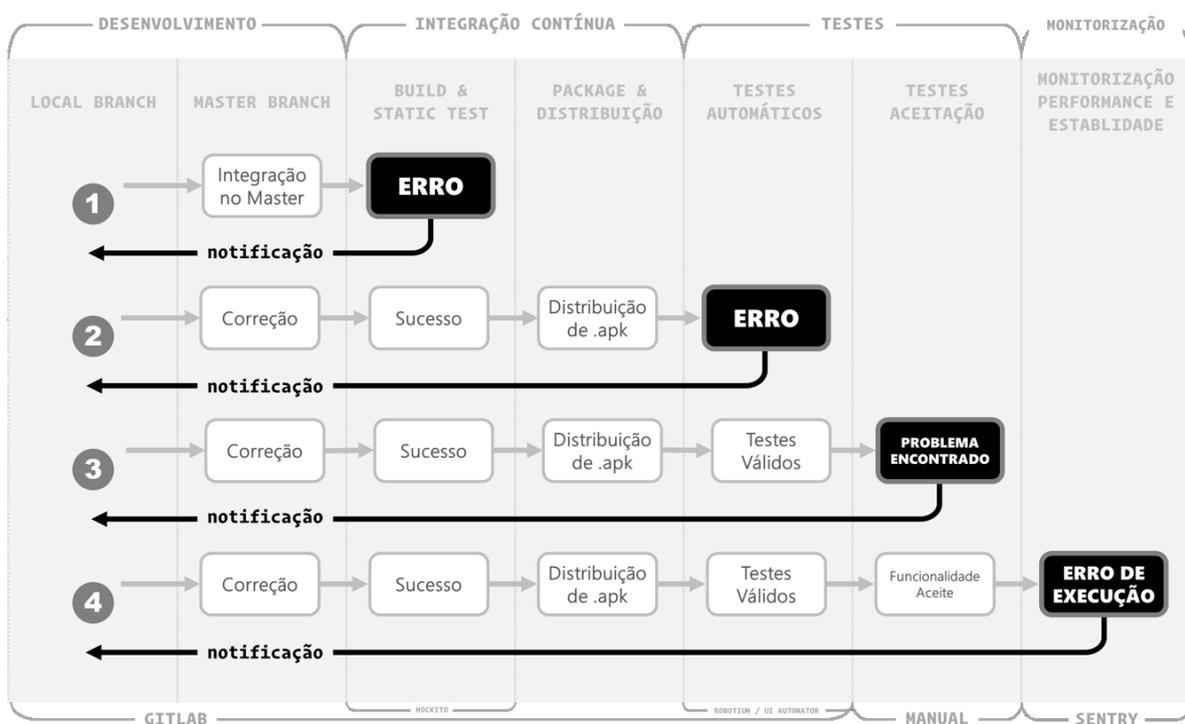


FIGURA 6-5 – PROCESSO DE DESENVOLVIMENTO E TESTE DE NOVAS FUNCIONALIDADES COM NOTIFICAÇÃO DE ERROS

Através da utilização de um conjunto de ferramentas foi possível automatizar quase todo o processo de desenvolvimento, integração, testes e monitorização de aplicações (Figura 6-5).

Tal como representado, o processo inicia-se como desenvolvimento local de novas funcionalidades e, após a validação da funcionalidade localmente, ela é fundida com o repositório de código principal (Figura 6-5 – ❶). Após a unificação do código fonte, será iniciado um processo automático que irá iniciar a compilação e a execução de teste de análise estática (*Lint* - <https://developer.android.com/studio/write/lint.html>, *PMD* - <https://pmd.github.io/>, *FindBug* - <http://findbugs.sourceforge.net/>) e unitários (*Mockito* - <http://site.mockito.org/>, *NUnit* - <https://www.nunit.org/>) do código fonte da versão principal sendo que, caso ocorra algum erro, este será transmitido ao utilizador, apresentando com detalhe o erro que ocorreu. Esta etapa tem como objetivos a manutenção do repositório sem erros de compilação, deteção de potenciais erros através de análises estáticas e, principalmente, evitar a existência de regressões ao repetir o conjunto de testes unitários para todas as versões de código.

Na situação em que ocorre algum erro nesta fase, esse será corrigido e, após nova submissão (Figura 6-5 – ❷), o processo de compilação e validação estática será repetido. Só após o sucesso deste, será preparado e assinado o pacote de instalação com a aplicação prosseguindo assim com a fase dos testes automáticos (*Robotium* - <http://www.robotium.org>, *UIAutomator* - <https://developer.android.com/training/testing/ui-testing/uiautomator-testing.html>, *Xamarin.UITest* – <https://developer.xamarin.com/guides/testcloud/uitest/>).

Caso ocorra algum erro este será registado com associação do output do processo para posterior análise e correção, sendo que o principal objetivo é a validação automática de potenciais regressões com as alterações ao código. Também se pretende neste processo testar os fluxos de interação, comunicação e adaptação em ambiente controlado, isto é, com contextos pré-definidos. Após a correção de algum erro que possa ter sido identificado nos testes automáticos, este será distribuído para o equipamento físico para a realização de testes de aceitação manuais. Relativamente aos equipamentos utilizados para os testes de aceitação manuais foram utilizados os seguintes dispositivos, Sony Xperia Z1 (Android stock 4.2.2 / 5.1, 2GB RAM), Samsung Galaxy S5 (Android stock 4.2.2 / 5.1, 2GB RAM).

Ao atingir esta etapa, a aplicação será testada em ambiente real para validação da funcionalidade pretendida onde, caso a funcionalidade não funcione como pretendido (Figura

6-5 – ③), será registado o problema com a descrição do que se pretendia e do comportamento atual. Nesta situação é realizado o teste nas salas de aula da Universidade de Trás-os-Montes e Alto Douro, avaliando os resultados nas mesmas condições dos alunos.

No entanto, mesmo que não ocorram erros e a funcionalidade tenha sido validada, a aplicação está sob monitorização permanente (Figura 6-5 – ④), onde todas as exceções e erros serão enviados, com detalhes, para uma plataforma de monitorização e onde qualquer erro ou comportamento indevido poderá ser analisado e corrigido, uma vez que estes ficaram associados à submissão de código, realizada na primeira etapa do processo.

Este processo, apesar de apresentar alguma complexidade, principalmente na fase inicial, onde é necessário configurar e personalizar a pipeline, apresenta vantagens significativas uma vez que, devido a automatização de todo o processo de teste e distribuição, é possível identificar erros, corrigi-los e colocar as novas correções nos dispositivos. Esta última necessidade é fundamental, especialmente quando se utiliza uma arquitetura heterogénea como a proposta, onde os serviços são independentes uns dos outros e, se por lapso, algum dos serviços não estiver atualizado, a aplicação pode gerar erros que não são reais e levar à perda desnecessária de tempo.

Assim, no desenvolvimento desta prova de conceito, o processo de desenvolvimento, aqui apresentado, foi replicado para cada um dos módulos independentes, uma vez que estes podem ser desenvolvidos em tecnologias diferentes (Java, C#), sob plataformas de desenvolvimento diferentes (Android SDK/Xamarin), com ferramentas de testes diferentes, e, mesmo assim, podemos beneficiar de um processo automatizado de desenvolvimento, distribuição e testes.

6.2.2. Caso de estudo – AlunosUTAD

Nesta tese, propõe-se a criação de um modelo contextual antecipatório que tem como principal objetivo a criação de um sistema que permita a reutilização da informação contextual inferida por várias aplicações. Este objetivo surge porque, na nossa visão, o principal obstáculo à utilização de contexto nas aplicações é a complexidade do processo em si, o que acrescentado

à dificuldade inerente ao processo de desenvolvimento aplicacional, torna a tarefa muitas vezes quase impossível.

Para validar a arquitetura proposta e, principalmente os pilares que a definem, reutilização, heterogeneidade e simplicidade de implementação, foi criada uma aplicação de exemplo, utilizando um problema real para melhorar com a utilização de contexto. Com esse objetivo foi desenvolvida, de raiz, uma aplicação para dar resposta aos problemas apresentados na secção 6.1 (pág. 137). Esta aplicação, foi inicialmente desenvolvida com os requisitos necessários mas, sem a utilização de contexto. Na Figura 6-6 são apresentados alguns *screenshots* da aplicação nesta fase.

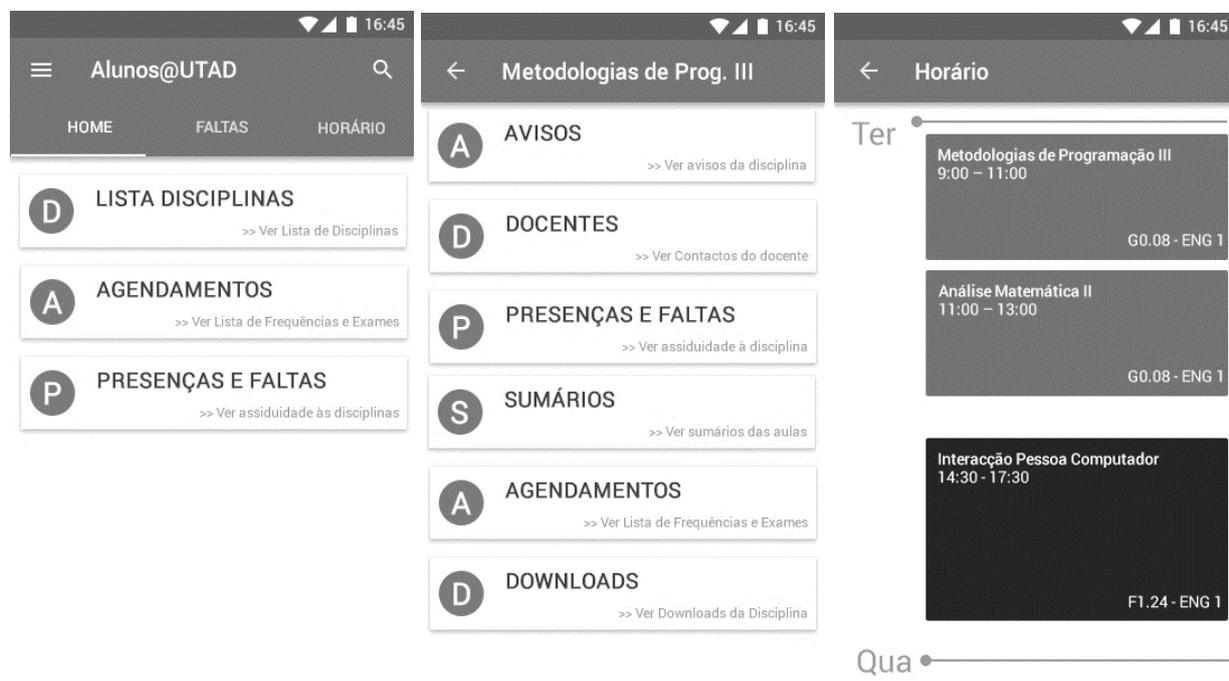


FIGURA 6-6 – ASPETO DA APLICAÇÃO ALUNOSUTAD SEM CAPACIDADE CONTEXTUAL

Após a conclusão desta etapa, procedeu-se à implementação de informação contextual com base nos requisitos contextuais (pág. 140), modelos de interação (pág.142) e fontes contextuais, apresentadas anteriormente (pág. 122 e 126). Com a implementação do consumo da informação contextual do utilizador, a aplicação apresenta uma interface que se atualiza consoante as necessidades ou cenários.

A aplicação apresenta a seguinte variação consoante este se encontra a caminho de uma aula, ou dentro da aula, sendo assim apresentada a informação que se considera mais importante, nomeadamente a localização da aula que é necessária quando este se encontra a caminho, mas não quando a aula está a decorrer (Figura 6-7).

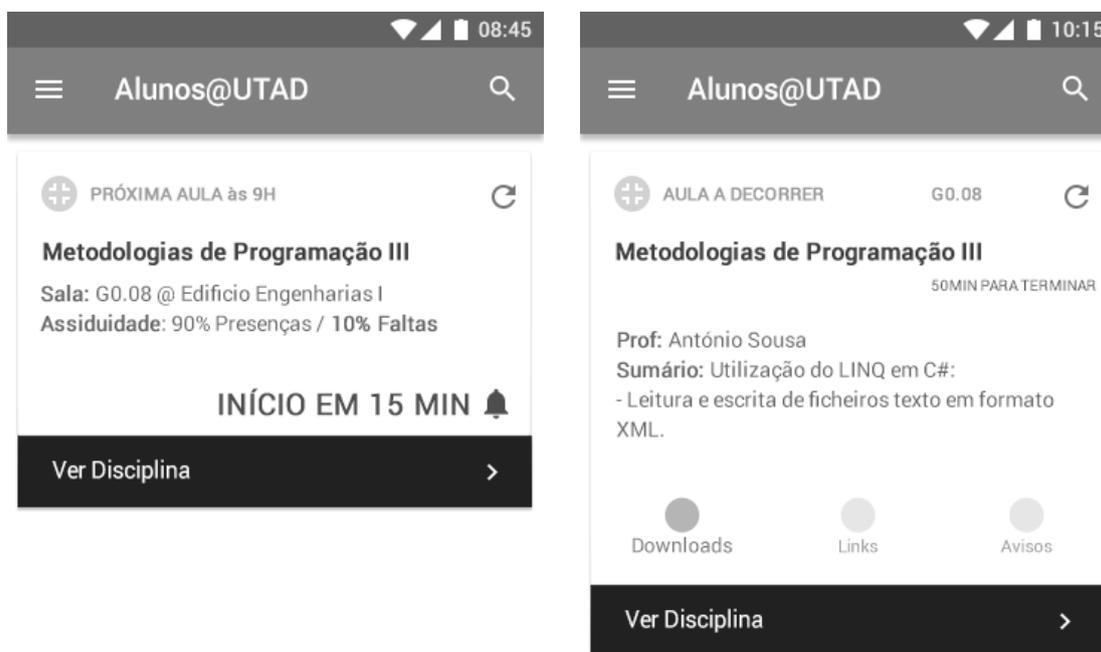


FIGURA 6-7 – ASPETO DA APLICAÇÃO ALUNOSUTAD COM CAPACIDADE CONTEXTUAL NAS SITUAÇÕES EM QUE SE ENCONTRA A CAMINHO DA AULA, E DENTRO DA DISCIPLINA

No entanto, a aplicação irá mudar para os detalhes da disciplina, apenas na situação em que o aluno se encontre dentro da sala, sendo que, caso ele esteja atrasado, será mantida a informação sobre a sala, mas com a variação relativa ao tempo de atraso (Figura 6-8).

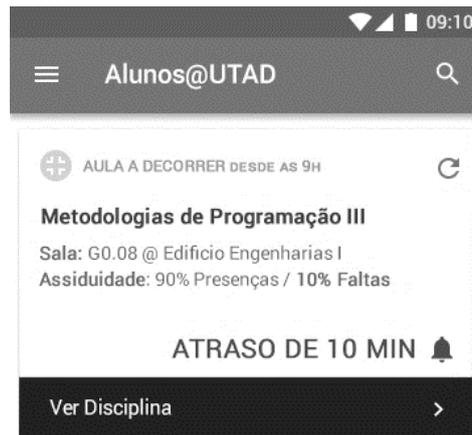


FIGURA 6-8 – ASPETO DA APLICAÇÃO ALUNOSUTAD COM CAPACIDADE CONTEXTUAL NA SITUAÇÃO EM QUE O ALUNO SE ENCONTRA ATRASADO

Em situações de transição, como a do fim de uma aula e início de outra em edifício diferente, a aplicação irá apresentar a seguinte informação (Figura 6-9).

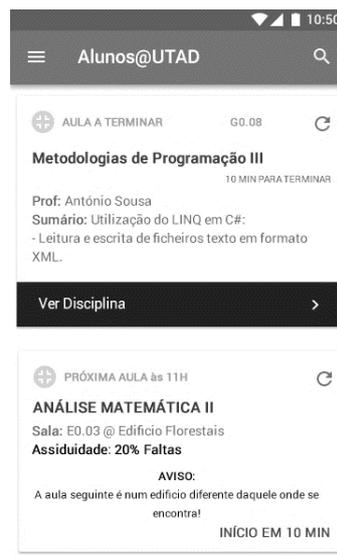


FIGURA 6-9 – ASPETO DA APLICAÇÃO ALUNOSUTAD EM SITUAÇÃO DE TRANSIÇÃO DE DISCIPLINA

Por fim, existem as situações de alertas onde o utilizador é alertado, de diversos modos, consoante a gravidade do problema, sendo dada a possibilidade de agendar um alarme, cuja antecipação irá variar se for a primeira aula (1h antes) ou intermédia (10min). O aspeto é representado na Figura 6-10.

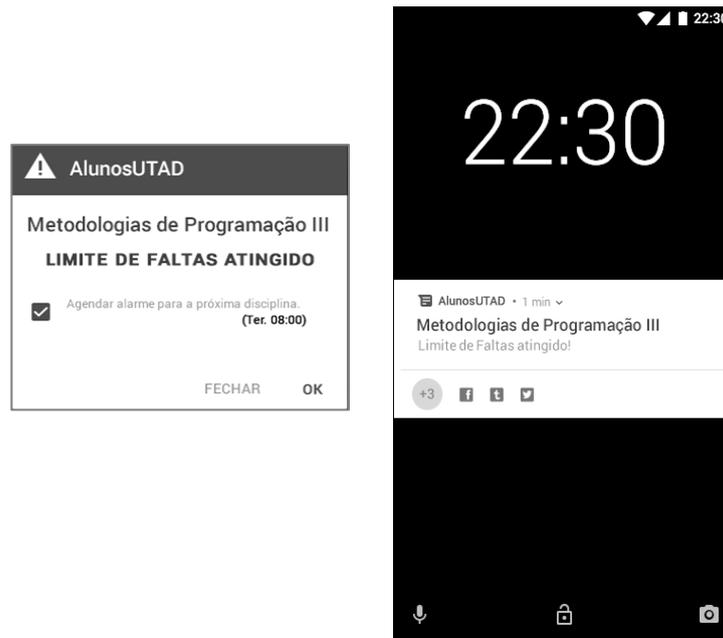


FIGURA 6-10 – ASPETO DAS NOTIFICAÇÕES ALUNOSUTAD EM SITUAÇÕES DE ALERTA IMPORTANTE

6.3. Validação do modelo

Como referido anteriormente, o âmbito desta tese não é a avaliação da adaptação, utilizando o contexto, mas sim a construção de um sistema que permite, a diversos programadores, implementar a informação contextual sem a complexidade inerente a essa implementação. O que se pretende era o desenvolvimento de um middleware que contém um conjunto alargado e heterogéneo de fontes contextuais, que podem ser consumidas por qualquer aplicação, sem que haja a necessidade de estas saberem como esta é implementada.

A validação do modelo em si, não é propriamente a aplicação AlunosUTAD, mas sim o processo de desenvolvimento desta, mais especificamente, a fase de implementação dos requisitos contextuais, uma vez que os obstáculos a que o ADAPT dá resposta são principalmente direcionados ao processo de desenvolvimento. Por este motivo, a realização de testes desta aplicação com utilizadores externos considerou-se desnecessária, uma vez que a avaliação desta incidiria mais sobre a avaliação da aplicação e a adaptação propriamente dita, do que sobre o ADAPT em si.

Contudo, de forma a poder validar se a informação estaria a ser corretamente inferida e se a arquitetura proposta apresentava vantagens, foi realizado um conjunto de testes que permitiram aferir se o sistema está a funcionar corretamente e conforme o pretendido. Desse modo, foram realizados 2 conjuntos de testes: um dedicado à avaliação da inferência contextual, visando a avaliação da interpretação dos sensores e das implementações dos algoritmos de categorização e inferência, sendo o principal destaque, a execução correta e sem erros nos vários dispositivos. Esta fase, foi realizada de forma automática na fase de *testes automáticos* do processo de desenvolvimento apresentado na Figura 6-5 (pág. 146).

A utilização de testes automáticos para a validação destes componentes do ADAPT, permitiu simular o comportamento e ações do utilizador em diversas situações, previamente planeadas, e verificar se os módulos se comportavam conforme o previsto, o que permitia detetar mais facilmente regressões quando eram realizadas alterações, ou implementação de novas funcionalidades.

Relativamente à fonte contextual de localização interna, implementada com recurso à aplicação da técnica *Random Forest*, de forma a ser possível realizar a implementação, foi necessário realizar o mapeamento prévio utilizando, para isso, o código apresentado na secção 6.4.2. O processo de recolha foi realizado de forma automática pela aplicação a cada 5 segundos, após a identificação onde as amostras iam ser recolhidas, sendo pedido, em seguida ao utilizador, que se movimenta após a recolha da amostra, sendo que, o recomendado é a recolha de, no mínimo, 100 amostras por divisão. Foi possível realizar o mapeamento de uma sala inteira em aproximadamente, 8-10 min, tendo sido recolhidas um total de quase 2000 amostras para a identificação de 12 salas.

Exemplo do registo obtido pela aplicação

```
F0.18,2d85d8ea-bz8d-4842-bb80-ded2840b9116,00:1c:f0:07:93:3c,Eduroam,-70,2412
F0.18,2d85d8ea-bz8d-4842-bb80-ded2840b9116,00:1c:f0:07:92:9c,Eduroam,-84,2462
F0.18,7b9cb38d-z67e-4c3b-9310-00fc876e5b5a,00:1c:f0:07:93:3c,Eduroam,-70,2412
F0.18,7b9cb38d-z67e-4c3b-9310-00fc876e5b5a,00:1c:f0:07:92:9c,Eduroam,-84,2462
F0.18,0abcf392-24s1-49df-9bae-9baf3798d428,00:1c:f0:07:93:3c,Eduroam,-70,2412
F0.18,0abcf392-24s1-49df-9bae-9baf3798d428,00:1c:f0:07:92:9c,Eduroam,-85,2462
F0.18,0abcf392-24s1-49df-9bae-9baf3798d428,2c:9e:5f:18:ee:37,Eduroam,-88,2437
F0.18,0abcf392-24s1-49df-9bae-9baf3798d428,94:cc:b9:03:e2:c1,Eduroam,-89,2442
F0.18,0abcf392-24s1-49df-9bae-9baf3798d428,00:26:5a:1f:93:0c,Eduroam,-97,2417
F0.18,468fecf8-3511-4915-bcca-14d9be99a5fb,00:1c:f0:07:93:3c,Eduroam,-70,2412
```

(...)

Como se pode verificar, no exemplo da recolha as redes WiFi, todas as amostras possuem o mesmo nome, o que é comum em ambientes que possuem a capacidade de roaming WiFi. No entanto, essa situação não é problemática para a implementação, uma vez que este não será utilizado pois, para identificar o *Access Point*, será utilizado o *BSSID* do mesmo. Dois dos parâmetros que são possíveis otimizar nesta técnica são: o número de árvores e a profundidade das mesmas. Assim, com o objetivo de identificar a melhor configuração, foram realizadas com variância de profundidade e tendo-se obtido os seguintes resultados:

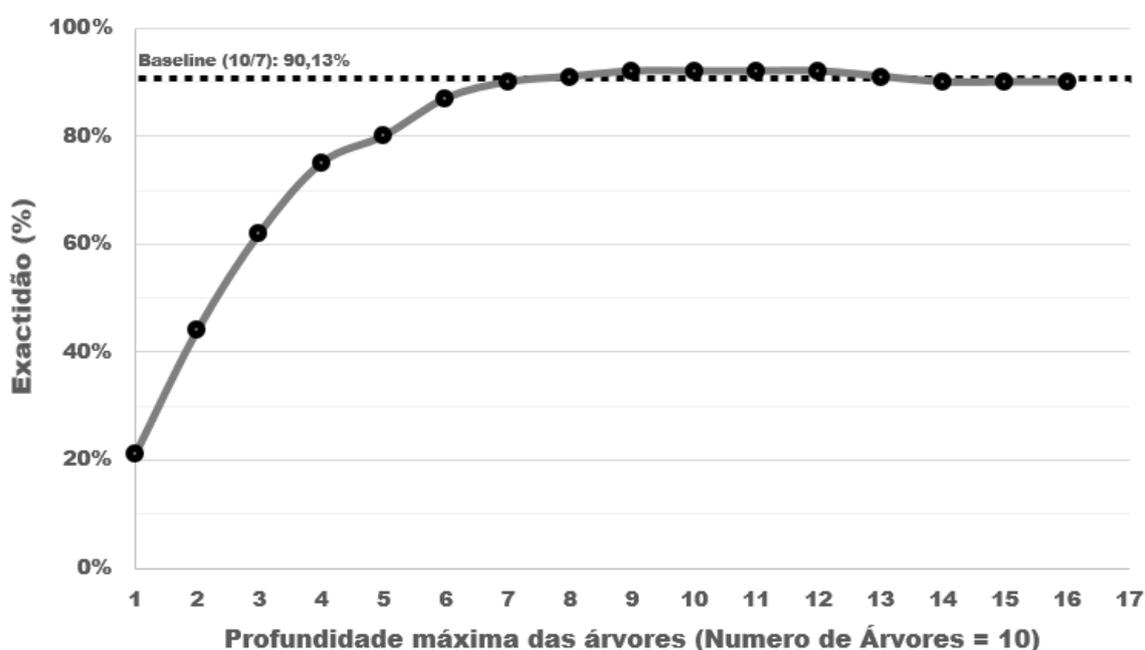


FIGURA 6-11 – PERFORMANCE OBTIDA PELA *RANDOM FOREST* COM VARIAÇÃO DO NÚMERO DE PROFUNDIDADE MÁXIMA E NÚMERO DE ÁRVORES = 10

Como base de comparação, os resultados foram confrontados com a execução, com os valores por defeito da biblioteca (10 árvores com profundidade máxima de 7 níveis), que corresponde a uma exatidão de 90,13%. A Figura 6-11, demonstra que a performance aumenta significativamente com a variação da profundidade da árvore entre os níveis 1-7, aumentando de forma mais reduzida até atingir o valor máximo no nível 9 (92,11%). Após este nível, o valor não melhora, chegando mesmo a piorar após o nível 12.

Em seguida, testou-se a variação em relação ao número de árvores da floresta, sendo novamente utilizado como base comparativa, os valores por defeito da biblioteca, e como valor de profundidade máxima, o melhor valor obtido anteriormente, 9 níveis.

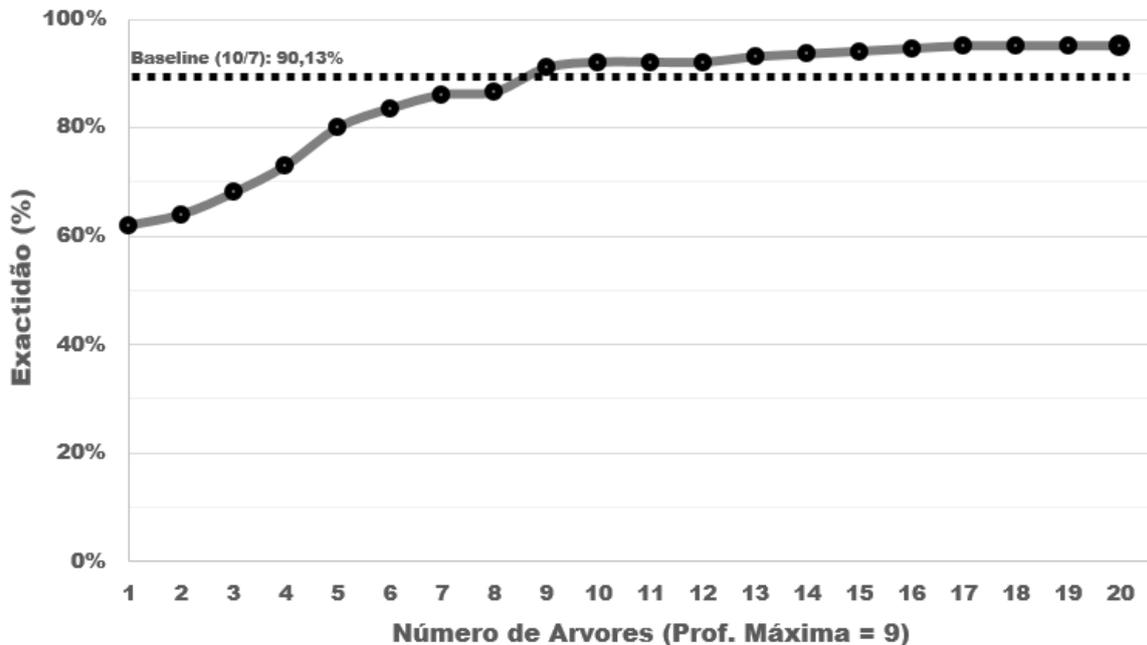


FIGURA 6-12 – PERFORMANCE OBTIDA PELA *RANDOM FOREST* COM VARIAÇÃO DO NÚMERO DE ÁRVORES, COM PROFUNDIDADE MÁXIMA = 9

De forma semelhante ao teste anterior, a Figura 6-12, demonstra que a performance melhora com o aumento de árvores na fase inicial, obtendo-se melhorias significativas até às 9 árvores, onde começa a ter um aumento mais reduzido, atingindo-se o máximo de 95% com 17 árvores. Esta situação acontece porque, em cada árvore da floresta, existe um ramo com essa informação e, quando maior for o número de árvores a votar, maior é a confiança obtida no resultado. No entanto, o tempo de treino necessário para esta configuração pode possuir um impacto significativo na performance do treino. Uma vez que o modelo não é treinado no dispositivo, mas sim num equipamento com maiores características sempre que for mapeada uma nova zona, este impacto não se revelou significativo para a nossa situação pois, a partir do momento em que este se encontrava treinado, a performance de uso não apresentou atrasos significativos.

Relativamente ao reconhecimento da atividade humana, implementados com uma CNN usando o *TensorFlow*, os resultados obtidos foram igualmente satisfatórios apresentando, em algumas das interações, os seguintes resultados apresentados na Tabela 6-4. No entanto, devido às limitações da metodologia de teste não foi possível obter outro tipo de informação, nomeadamente, quantificação do número de falsos positivos, uma vez que apenas as situações corretas foram consideradas válidas.

TABELA 6-4 – RESULTADO DOS TESTES DE RECONHECIMENTO DE ATIVIDADE HUMANA COM UMA CNN

ATIVIDADE	IDENTIFICAÇÃO CORRECTA (%)
CORRIDA	96,12%
CAMINHAR	97,14%
SENTADO	90,18%
EM PÉ	96,72%
DE CARRO	98,45%

Como referido, para este caso não foram recolhidos dados para realizar o treino da CNN de reconhecimento de atividade, tendo sido utilizado o *dataset actitracker* de Lockhart et al. (2011) para o treino e validação com testes automáticos.

Após a passagem com sucesso da etapa anterior, a aplicação procedia para a segunda fase e tinha como objetivo a avaliação da qualidade da inferência contextual, permitindo, assim, validar se o contexto inferido estava correto e se a adaptação realizada foi bem-sucedida ou não correspondendo, assim, fase de *testes de aceitação* do modelo apresentado na Figura 6-5 (pág. 146). Esta fase foi realizada através de avaliações manuais em ambiente real, permitindo verificar se os modelos implementados correspondiam à realidade e as inferências eram realizadas corretamente e, mais importante, se a aplicação conseguia utilizá-las para melhorar a sua experiência de utilização. Os testes realizados seguiram o guião apresentado pela Figura 6-13, onde cada etapa era testada fisicamente e em ambiente real, simulando as atividades do utilizador.

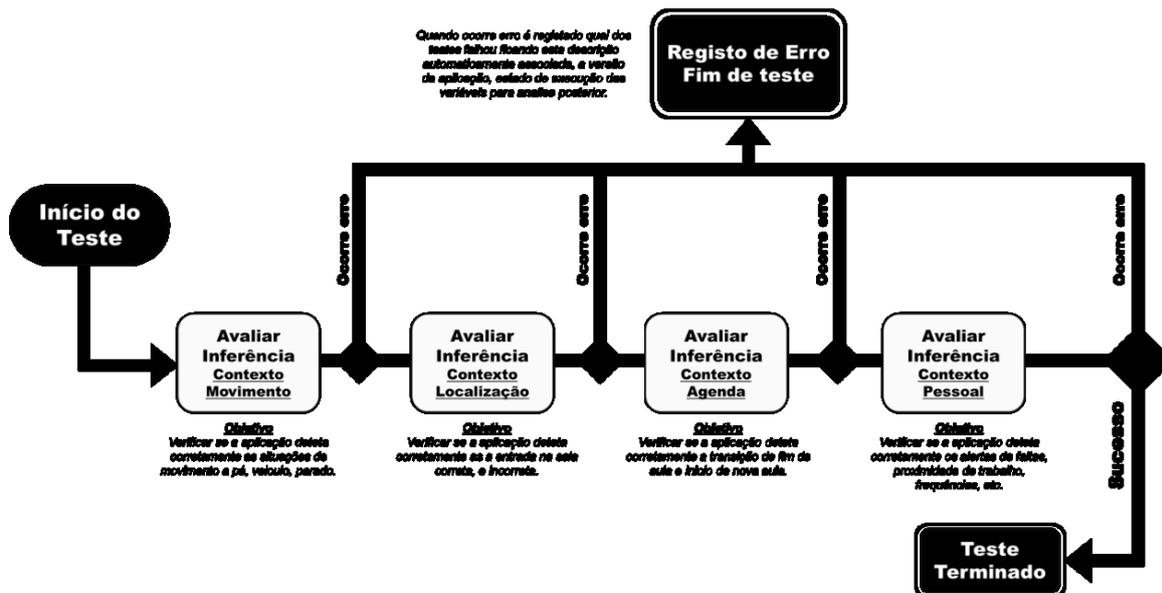


FIGURA 6-13 – GUIÃO DE AVALIAÇÃO DE TESTES DE ACEITAÇÃO DA APLICAÇÃO ALUNOSUTAD

Como representado na Figura 6-13, sempre que ocorria algum erro, ou algum dos testes não era válido, era registado o erro, com a descrição e a associação da informação contextual nesse momento, para análise e correção posterior. Com base neste processo, obtiveram-se os resultados apresentados na Tabela 6-5, ocorrendo situação semelhante ao caso anterior, onde apenas os casos válidos eram considerados, não sendo possível quantificar informação adicional, como falsos-positivos.

TABELA 6-5 – RESULTADO DOS TESTES DE MANUAIS DE ACEITAÇÃO COM BASE NO GUIÃO PROPOSTO

ETAPA	IDENTIFICAÇÃO CORRECTA (%)
CONTEXTO MOVIMENTO	94,12%
CONTEXTO LOCALIZAÇÃO	98,61%
CONTEXTO AGENDA	99,90%
CONTEXTO PESSOAL	99,90%

Como é possível verificar, a percentagem de sucesso obtida nos diversos testes é elevada, o que demonstra o potencial da utilização de contexto nas aplicações. Para o contexto de movimento, os principais problemas prendiam-se com a identificação de alguns movimentos, quando estes não existiam, nomeadamente em situações como tirar o telemóvel do bolso. Esta situação foi resolvida com a diminuição da taxa de atualização, o que permitiu suavizar a

variabilidade. Relativamente à identificação de localização interna, os principais problemas ocorriam na proximidade aos limites das salas.

É de notar que o caso de estudo apresenta algumas limitações que devem ser consideradas. A aplicação desenvolvida apenas implementa um conjunto de fontes contextuais reduzido, pelo que não foi possível inferir a capacidade de controlo e gestão do ADAPT em situações com um número elevado de fontes contextuais em funcionamento. Contudo, as fontes contextuais utilizadas foram selecionadas como forma de comprovação da capacidade de utilizar tecnologias diferentes e a capacidade de usar bibliotecas mais avançadas, como o *TensorFlow*, realizando o treino em sistemas dedicados e exportar o modelo para a aplicação móvel e, assim, poder beneficiar das capacidades deles.

Uma dificuldade na implementação das fontes contextuais, em arquitetura de micro serviços, é que a plataforma android requer, por questões de segurança, que todos os serviços que pretendam ser invocados automaticamente, tenham que ter sido iniciados pelo menos uma vez pelo utilizador, o que obriga à existência de uma interface por cada serviço.

Relativamente às fontes contextuais, nomeadamente à localização interna, requer previamente uma recolha de amostras para mapeamento do edifício a reconhecer. Isto pode-se tornar um problema, à medida que vai aumentando este número. Uma solução, pode passar por carregar dinamicamente o modelo de localização interna, apenas quando se encontra na proximidade desse edifício, tornando deste modo a própria fonte contextual sensível ao contexto.

Assim, e de forma resumida, os resultados dos testes realizados demonstraram a viabilidade e mais-valia que um sistema como o ADAPT mostrou ter acrescentando ao processo de desenvolvimento de aplicações móveis sensíveis ao contexto. Permitindo de forma simples incorporar a capacidade de usar determinada informação contextual sem saber propriamente como esta é a interpretação e inferência da mesma é implementada permite reduzir o tempo de desenvolvimento necessário.

7. CONCLUSÕES E TRABALHO FUTURO

Neste capítulo final, será apresentada uma síntese dos pontos mais relevantes do trabalho desenvolvido, sendo realizada uma análise crítica às características do modelo proposto e a avaliação do estado atual ADAPT. Serão tecidas algumas conclusões do trabalho com particular destaque ao nível da validação dos objetivos, propostos inicialmente. Serão também exploradas potenciais linhas de investigação futura, de forma a permitir a continuidade do trabalho realizado.

7.1. Conclusões

O objetivo principal desta tese é a definição de um modelo para a sensorização contextual e adaptação antecipatória que sirva como base para o desenvolvimento de uma arquitetura de software dinâmica de um sistema de partilha de informação contextual. Como apresentado nesta tese, o modelo contextual antecipatório, implementado pela arquitetura de suporte ao modelo, constitui uma solução para a problemática existente, classificando e inferindo a partir dos dados sensoriais, conhecimento relevante à ação do utilizador.

A validação do modelo contextual antecipatório e da arquitetura implementada, foi efetuada com base em experiências realizadas em situações reais usando, para o efeito, uma aplicação construída de raiz, onde a utilização de informação contextual pode apresentar-se uma mais-valia para os utilizadores.

O caso de estudo realizado visou a validação do modelo e da arquitetura, propostos nesta tese e também a validação ao nível das estruturas de dados e sistemas de partilha de informação contextual. No sentido de atingir o objetivo principal, apresentado no Capítulo 1, foram realizados os seguintes objetivos parcelares.

- Realizou-se um estudo sobre o contexto, a sensibilidade ao contexto e ciclo de vida contextual que identificou os seus principais requisitos e características. Neste estudo também foram identificados os vários domínios do contexto, o que nos permitiu compreender melhor a multiplicidade de informação que pode ser relevante para a contextualização do utilizador.
- Realizou-se um estudo sobre o estado da arte das arquiteturas de *middleware*, de metodologias de comunicação e abstração de sensores para a criação de serviços isolados e independentes, tendo sido identificados os trabalhos relevantes.
- Efetuou-se a avaliação do papel das arquiteturas de *middleware* contextual no desenvolvimento de aplicações sensíveis ao contexto. Neste âmbito, foram

avaliadas diversas propostas arquiteturais e validadas as suas vantagens e desvantagens.

- Propôs-se um modelo inovador para a construção de um *middleware*, que deu resposta ao propósito da tese: responder às necessidades de sensibilidade contextual permitindo, assim, o uso desta informação para a adaptação dinâmica e antecipatória por parte das aplicações.
- Pesquisaram-se implementações de tecnologias que possam servir de suporte à implementação de um *middleware* de inferência contextual antecipatória baseada no modelo proposto, entre ferramentas comerciais, livres e de código aberto. Foram consideradas as necessidades dos diversos módulos funcionais do modelo proposto.
- A arquitetura especificada nesta tese foi implementada num protótipo, construído para a validação da mesma, para utilização por utilizadores externos.
- Foi realizada a avaliação qualitativa da arquitetura desenvolvida, que permitiu assegurar que as funcionalidades implementadas seguissem as especificações definidas, bem como se assegurava o bom funcionamento da aplicação.

Deste modo, os objetivos parcelares, focados no capítulo introdutório, foram atingidos com sucesso, tendo sido apresentadas contribuições inovadoras na área de sensorização contextual e adaptação dinâmica e antecipatória.

Ao nível das questões de investigação, o caso de uso implementado permitiu-nos dar resposta às mesmas: através da identificação das principais dificuldades da interação com o sistema (RQ-1); reconhecer o contexto relevante do utilizador, nomeadamente detetar o seu tipo de atividade, e a sua localização dentro do edifício com o uso dos sensores internos do equipamento móvel (RQ-2); com base no seu contexto, adaptar a interface da aplicação de forma a permitir uma melhor experiência de utilização e, principalmente, obter a informação pretendida mais rapidamente (RQ-3); através da arquitetura proposta pelo sistema ADAPT, permitir a partilha da informação contextual por várias aplicações, como apresentado nesta tese, o tipo de

atividade que é consumido pela aplicação e pela fonte contextual de localização, e pelo módulo de gestão energética, o que permite reduzir a necessidade de recalculando este contexto nas aplicações que o pretendiam usar (RQ-4); Através do uso de uma arquitetura de micro serviços isolados, promoveu-se a capacidade de reutilização e simplicidade de integração através da redução de dependências, sendo que as aplicações apenas necessitam de saber o que precisam, e não quem o pode inferir ou como é que se pode inferir (RQ-5).

A privacidade do utilizador é, no entanto, uma preocupação que deve ser tida em conta aquando o desenvolvimento de sistemas sensíveis ao contexto, uma vez que, se tem acesso a um conjunto de informação privada do utilizador. Um exemplo do cuidado a ter com esse tipo de informação, por exemplo no sistema AlunosUTAD, tem a ver com a monitorização da localização do aluno, dentro e fora de edifícios. No caso da nossa aplicação, uma das decisões arquiteturais restringiu a utilização do sistema apenas em modo local, ou seja, nenhum tipo de informação era transmitido para servidores externos, de forma a poder salvaguardar a privacidade do utilizador. No entanto, esta temática tem que ser tida em conta no desenvolvimento destes sistemas, uma vez que o potencial da utilização e o valor da informação contextual terá que ser equilibrado com o direito à privacidade de cada utilizador, sendo necessário que no processo de desenvolvimento de fontes contextuais se tenha em conta a anonimização dos dados.

7.2. Linhas de orientação futura

O trabalho apresentado nesta tese, apesar de ser completo por si só, não esgota totalmente as potencialidades do tema, tal como o modelo proposto, que não representa uma solução definitiva para o problema, mas antes uma contribuição valiosa para o debate da temática e do problema na comunidade científica. Assim, são apresentadas em seguida algumas linhas de orientação futura que ambicionam inspirar a continuidade desta investigação. Um dos pontos mais frágeis desta proposta é a realização do processamento no dispositivo móvel, pelo que o sistema apresenta algumas limitações ao nível das capacidades do dispositivo onde este é executado. Neste contexto, a exploração de técnicas para delegar o processamento para outros

equipamentos ou servidores com maiores capacidades técnicas e algorítmicas pode constituir uma linha de trabalho futuro, com elevado potencial. Um grande potencial, relativo a este obstáculo, é apresentar-se com a evolução das redes de comunicação de nova geração, nomeadamente ao nível do 5G e a capacidade de este poder enviar necessidades de processamento para as estações para calcular e devolverem.

Outra possível linha de investigação passa pela utilização das adaptações contextuais e antecipatórias em sistemas de apoio a navegação para cegos, uma vez que uma parte das simplificações e aumento da eficiência que a adaptação contextual permite, pode tornar-se benéfico também para pessoas com alguma limitação ou deficiência.

O protótipo construído para a validação da arquitetura e do modelo contextual antecipatório, pretende apenas demonstrar as capacidades do sistema, apresentando algumas limitações em relação às funcionalidades como aplicação móvel para alunos. Deste modo, o desenvolvimento de novas funcionalidades poderia permitir a continuidade da utilização desta aplicação por parte dos alunos, mas também, garantir o melhoramento das técnicas de inferência contextual, através da continuidade de validação de conceitos.

8. LISTA DE PUBLICAÇÕES ORIGINAIS

Durante os trabalhos relacionados com esta tese de doutoramento, foram publicados os seguintes artigos originais.

José Martins, João Barroso, Ramiro Gonçalves, André Sousa, Miguel Bacelar, Hugo Paredes.

(2015). Transforming e-Procurement Platforms for PEPPOL and WCAG 2.0

Compliance – The anoGov-PEPPOL Project. *Information Science and Applications*, 339(4), pp. 973-980. doi:10.1007/978-3-662-46578-3_116

Hugo Paredes, Hugo Fernandes, André Sousa, Renata Fortes, Fernando Koch, Vitor Filipe,

João Barroso. (2015). CanIHelp: A Platform for Inclusive Collaboration, *Universal Access in Human-Computer Interaction - Access to Interaction 9176*. pp. 474-483.

doi:10.1007/978-3-319-20681-3_45

- Hugo Fernandes, André Sousa, Hugo Paredes, Vitor Filipe, João Barroso. (2015). Feature Detection Applied to Context-Aware Blind Guidance Support. *Universal Access in Human-Computer Interaction - Access to the Human Environment and Culture 9178*. pp. 129-138. doi:10.1007/978-3-319-20687-5_13
- Hugo Paredes, Hugo Fernandes, André Sousa, Luís Fernandes, Fernando Koch, Renata Fortes, Vítor Filipe, João Barroso. (2015). Exploring Smart Environments Through Human Computation for Enhancing Blind Navigation. *Advances in Social Computing and Multiagent Systems 541*. pp. 66-76. doi:10.1007/978-3-319-24804-2_5
- André Sousa, João Barroso, Hugo Paredes, Hugo Fernandes, Vítor Filipe. (2015). Context-aware, Accessibility and Dynamic Adaptation of Mobile Interfaces in Business Environments, *Procedia Computer Science 67*. pp. 397-402. doi:10.1016/j.procs.2015.09.284
- Hugo Paredes, Hugo Fernandes, André Sousa, Fernando Koch, Vitor Filipe, João Barroso. (2016). Pervasive Crowd Mapping for Dynamic Environments. *Advances in Computer Science Research*. pp.157-162. ISBN 978-94-6252-146-9. - *Best Paper Award*
- Reis, A., Martins, P., Borges, J., Sousa, A., Rocha, T., Barroso, J. (2017). Supporting Accessibility in Higher Education Information Systems: a 2016 Update. *Aceite HCI International 2017* URL: (publication pending).

9. BIBLIOGRAFIA

- Abowd, G. D. (1999). Software engineering issues for ubiquitous computing. *Proceedings of the 21st international conference on Software engineering*, 75-84.
doi:10.1145/302405.302454
- Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness. *Handheld and Ubiquitous Computing, 1707*, 304-307. doi:10.1007/3-540-48157-5_29
- Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A., & Hopper, A. (2001). Implementing a sentient computing system. *Computer*, 34(8), 50-56.
doi:10.1109/2.940013
- Aharony, N., Pan, W., Ip, C., Khayal, I., & Pentland, A. (2011). The social fmri. *Proceedings of the 13th international conference on Ubiquitous computing - UbiComp '11*, 445.
doi:10.1145/2030112.2030171
- Ahn, S., & Kim, D. (2006). Proactive context-aware sensor networks. In K. Römer, H. Karl, & F. Mattern (Eds.), *Wireless sensor networks: Third european workshop, ewsn 2006, zurich, switzerland, february 13-15, 2006. Proceedings* (pp. 38-53). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Aiello, F., Fortino, G., Gravina, R., & Guerrieri, A. (2011). A java-based agent platform for programming wireless sensor networks. *The Computer Journal*, 54(3), 439-454.
doi:10.1093/comjnl/bxq019

- Aiken, R. J., Strassner, J., Carpenter, B. E., Foster, I., Lynch, D. C. A., Mambretti, J., Moore, D. R. W., & Teitelbaum, B. R. (2000). *Rfc 2768 : Network policy and services: A report of a workshop on middleware*. Retrieved from <https://www.ietf.org/rfc/rfc2768.txt>
- Akman, V., & Surav, M. (1997). The use of situation theory in context modeling. *Computational Intelligence, 13*(3). Retrieved from <http://cogprints.org/460/>
- Ameyed, D., Miraoui, M., & Tadj, C. (2011). A survey of prediction approach in pervasive computing. *International Journal of Scientific and Engineering Research, 6*(5). Retrieved from <http://www.ijser.org/researchpaper%5CA-Survey-of-Prediction-Approach-in-Pervasive-Computing.pdf>
- Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2012). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In J. Bravo, R. Hervás, & M. Rodríguez (Eds.), *Ambient assisted living and home care: 4th international workshop, iwaal 2012, vitoria-gasteiz, spain, december 3-5, 2012. Proceedings* (pp. 216-223). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Bak, P., Melamed, R., Moshkovich, D., Nardi, Y., Ship, H., & Yaeli, A. (2015, June 27 2015- July 2 2015). *Location and context-based microservices for mobile and internet of things workloads*. Paper presented at the 2015 IEEE International Conference on Mobile Services.
- Barbosa, L. F., Alves, P., & Barroso, J. (2011, 15-18 June 2011). *Side - teaching support information system*. Paper presented at the 6th Iberian Conference on Information Systems and Technologies (CISTI 2011).
- Barwise, J., & Perry, J. (1981). Situations and attitudes. *Attitudes & Social Adaptation, 93*. doi:10.1016/b978-0-08-026074-7.50013-6
- Bauer, J., Kutsche, R. D., & Ehrmantraut, R. (2003). *Identification and modeling of contexts for different information scenarios in air traffic*. (Ph.D). Retrieved from <https://pdfs.semanticscholar.org/3fe5/ead548ca4bf9fffe8a6032dd741275b1573.pdf>

- Bazire, M., & Brézillon, P. (2005). Understanding context before using it. In A. Dey, B. Kokinov, D. Leake, & R. Turner (Eds.), *Modeling and using context: 5th international and interdisciplinary conference context 2005, paris, france, july 5-8, 2005. Proceedings* (pp. 29-40). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Beigl, M., Gellersen, H.-W., & Schmidt, A. (1999). There is more to context than location. *Computers & Graphics*, 23, 893-901. doi:10.1016/S0097-8493(99)00120-X
- Bellavista, P., Corradi, A., Fanelli, M., & Foschini, L. (2012). A survey of context data distribution for mobile ubiquitous systems. *ACM Computing Surveys*, 44(4), 1-45. doi:10.1145/2333112.2333119
- Bellotti, V., & Edwards, K. (2001). Intelligibility and accountability: Human considerations in context-aware systems. *Human-Computer Interaction*, 16, 193-212. Retrieved from http://www.tandfonline.com/doi/abs/10.1207/S15327051HCI16234_05
- Biegel, G., & Cahill, V. (2004, 14-17 March 2004). *A framework for developing mobile, context-aware applications*. Paper presented at the Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the.
- Boashash, B. (1992). Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals. *Proceedings of the IEEE*, 80(4), 520-538. doi:10.1109/5.135376
- Bobillo, F., & Straccia, U. (2011). Fuzzy ontology representation using owl 2. *International Journal of Approximate Reasoning*, 52(7), 1073-1094. doi:10.1016/j.ijar.2011.05.003
- Böhmer, M., Hecht, B., Schöning, J., Krüger, A., & Bauer, G. (2011). Falling asleep with angry birds, facebook and kindle. *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '11*, 47. doi:10.1145/2037373.2037383
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. doi:10.1023/a:1010933404324

- Brown, P. J., Chen, X., & Bovey, J. D. (1997). Context-aware applications: From the laboratory to the marketplace. *IEEE Personal Communications*, 4, 58-64.
doi:10.1109/98.626984
- Brunette, W., Sodt, R., Chaudhri, R., Goel, M., Falcone, M., Van Orden, J., & Borriello, G. (2012). Open data kit sensors. *Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys '12*, 351.
doi:10.1145/2307636.2307669
- Buchholz, T., Hochstatter, I., & Linnhoff-Popien, C. (2007). Distribution strategies for the contextualized mobile internet. *Electronic Commerce Research and Applications*, 6(1), 40-52. doi:10.1016/j.elerap.2006.04.002
- Buchholz, T., & Schiffers, M. (2003). Quality of context: What it is and why we need it *Proceedings of the 10th Workshop of the OpenView University Association: OVUA '03*. doi:10.1.1.147.565
- Calderoni, L., Ferrara, M., Franco, A., & Maio, D. (2015). Indoor localization in a hospital environment using random forest classifiers. *Expert Systems with Applications*, 42(1), 125-134. doi:10.1016/j.eswa.2014.07.042
- Capra, L., Emmerich, W., & Mascolo, C. (2001). Reflective middleware solutions for context-aware applications. In A. Yonezawa & S. Matsuoka (Eds.), *Metalevel architectures and separation of crosscutting concerns: Third international conference, reflection 2001 kyoto, japan, september 25–28, 2001 proceedings* (pp. 126-133). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Cassens, J., & Kofod, A. (2006). Using activity theory to model context awareness: A qualitative case study. *FLAIRS Conference*. Retrieved from <http://www.aaai.org/Papers/FLAIRS/2006/Flairs06-122.pdf>
- Castelli, G., Rosi, A., Mamei, M., & Zambonelli, F. (2007). A simple model and infrastructure for context-aware browsing of the world. *Fifth Annual IEEE International Conference on Pervasive Computing and Communications*, 229-238.
doi:10.1109/percom.2007.4

- CCS Insight. (2016, Jan 01). Smartphone sales to peak in western markets in 2017 as they enter new phase of maturity. <http://www.ccsinsight.com/press/company-news/2183-smartphone-sales-to-peak-in-western-markets-in-2017-as-they-enter-new-phase-of-maturity>.
- Chen, H. (2003). *An intelligent broker architecture for context-aware systems*. (Ph.D), University of Maryland, Baltimore County. Retrieved from <http://ebiquity.umbc.edu/paper/html/id/212/An-Intelligent-Broker-Architecture-for-Pervasive-Context-Aware-Systems>
- Cheverst, K., Mitchell, K., & Davies, N. (1999). Design of an object model for a context sensitive tourist guide. *Computers & Graphics*, 23(6), 883-891. doi:10.1016/s0097-8493(99)00119-3
- Chtcherbina, E., & Franz, M. (2003). Peer-to-peer coordination framework (p2pc): Enabler of mobile ad-hoc networking for medicine, business, and entertainment. *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet*. doi:10.1.1.11.8739
- CLIPS. (1985, Jan 01). Clips: A tool for building expert systems. <http://clipsrules.sourceforge.net/>.
- Crowley, J. L., Coutaz, J., Rey, G., & Reignier, P. (2002). Perceptual components for context aware computing. In G. Borriello & L. E. Holmquist (Eds.), *Ubicomp 2002: Ubiquitous computing: 4th international conference göteborg, sweden, september 29 – october 1, 2002 proceedings* (Vol. 2498, pp. 117-134). Berlin, Heidelberg: Springer Berlin Heidelberg.
- de Ipiña, D. L., & Katsiri, E. (2001). An eca rule-matching service for simpler development of reactive applications. *IEEE Distributed Systems Online*, 2, 1-7. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.7050&rep=rep1&type=pdf>

- De Pessemier, T., Dooms, S., & Martens, L. (2013). Context-aware recommendations through context and activity recognition in a mobile environment. *Multimedia Tools and Applications*, 72(3), 2925-2948. doi:10.1007/s11042-013-1582-x
- Devlin, K. (2005). *Confronting context effects in intelligence analysis: How can mathematics help?* Center for the Study of Language and Information. Stanford University. Retrieved from https://web.stanford.edu/~kdevlin/Papers/Context_in_Reasoning.pdf
- Dey, A., Abowd, G., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2), 97-166. doi:10.1207/s15327051hci16234_02
- Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1), 4-7. doi:10.1007/s007790170019
- Dey, A. K., Futakawa, M., Salber, D., & Abowd, G. D. (1999). *The conference assistant: Combining context-awareness with wearable computing*. Paper presented at the Proceedings of the 3rd IEEE International Symposium on Wearable Computers.
- Dey, A. K., Wood, A., & Abowd, G. D. (1998). Cyberdesk: A framework for providing self-integrating context-aware services. *Knowledge-Based Systems*, 11(1), 3-13. doi:10.1016/S0950-7051(98)00053-7
- Di Sciuillo, A.-M. (1999). Formal context and morphological analysis. *Context 99*, 1688, 105-118. doi:10.1007/3-540-48315-2_9
- Dickerson, R. F., Gorlin, E. I., & Stankovic, J. A. (2011). Empath. *Proceedings of the 2nd Conference on Wireless Health - WH '11*, 1. doi:10.1145/2077546.2077552
- Dijk, A. T. V. (2008). *Context and discourse*: Universitat Pompeu Fabra.
- Dourish, P. (2004). What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1), 19-30. doi:10.1007/s00779-003-0253-8
- Dustdar, S., Baldauf, M., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4), 263-277.

- Retrieved from
<http://www.inderscienceonline.com/doi/abs/10.1504/IJAHUC.2007.014070>
- Ekbia, H. R., & Maguitman, A. G. (2001). Context and relevance: A pragmatic approach. In V. Akman, P. Bouquet, R. Thomason, & R. Young (Eds.), *Modeling and using context: Third international and interdisciplinary conference, context 2001 dundee, uk, july 27–30, 2001 proceedings* (Vol. 2116, pp. 156-169). Berlin, Heidelberg: Springer.
- Evans, E. (2004). *Domain-driven design : Tackling complexity in the heart of software*. Boston: Addison-Wesley.
- Falaki, H., Mahajan, R., & Estrin, D. (2011). Systemsens. *Proceedings of the sixth international workshop on MobiArch - MobiArch '11*, 25.
doi:10.1145/1999916.1999923
- Feng, Y., Bagheri, E., Ensan, F., & Jovanovic, J. (2017). The state of the art in semantic relatedness: A framework for comparison. *The Knowledge Engineering Review*, 32.
doi:10.1017/S0269888917000029
- Fernandes, H., Faria, J., Martins, P., Paredes, H., & Barroso, J. (2013). Rfid mesh network as an infrastructure for location based services for the blind. In M. Kurosu (Ed.), *Human-computer interaction. Towards intelligent and implicit interaction: 15th international conference, hci international 2013, las vegas, nv, USA, july 21-26, 2013, proceedings, part v* (pp. 39-45). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Fernandes, H., Filipe, V., Costa, P., & Barroso, J. (2014). Location based services for the blind supported by rfid technology. *Procedia Computer Science*, 27, 2-8.
doi:10.1016/j.procs.2014.02.002
- Ferreira, D. (2013). *Aware: A mobile context instrumentation middleware to collaboratively understand human behavior*. (Ph.D), University of Oulu Retrieved from
<http://urn.fi/urn:isbn:9789526201900>

- Ferreira, D., Kostakos, V., & Dey, A. K. (2015). Aware: Mobile context instrumentation framework. *Frontiers in ICT*, 2. doi:10.3389/fict.2015.00006
- Find Project. (2016). Wifi fingerprint visualization [online image].
<https://www.internalpositioning.com/faq/>.
- Fowler, M. (2004). *Inversion of control containers and the dependency injection pattern - using a service locator*. Retrieved from
<https://martinfowler.com/articles/injection.html#UsingAServiceLocator>
- Fowler, M. (2015). *Microservices resource guide*. Retrieved from
<https://martinfowler.com/microservices/>
- Fowler, M., & Lewis, J. (2014). *Microservices: A definition of a new architectural term*. White Paper. Retrieved from <http://martinfowler.com/articles/microservices.html>,
- Fowler, M., & Parsons, R. (2011). *Domain-specific languages*. Upper Saddle River, NJ: Addison-Wesley.
- Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007). Myexperience. *Proceedings of the 5th international conference on Mobile systems, applications and services - MobiSys '07*, 57. doi:10.1145/1247660.1247670
- Gartner Inc. (2016, Jun 01). Gartner says worldwide smartphone sales to slow in 2016.
<http://www.gartner.com/newsroom/id/3339019>.
- Ginger Inc. (2016, Sep 01). Ginger.Io. <https://www.ginger.io>.
- Giunchiglia, F., Maltese, V., & Dutta, B. (2012). Domains and context: First steps towards managing diversity in knowledge. *Web Semantics: Science, Services and Agents on the World Wide Web*, 12-13, 53-63. doi:10.1016/j.websem.2011.11.007
- Gonçalves, B., Filho, J. P., & Andreão, R. V. (2008). Ecgaware: An ecg markup language for ambulatory telemonitoring and decision making support. *Proceedings of the First International Conference on Health Informatics*, 37-43.
doi:10.5220/0001037200370043

- Google inc. (2017). Android sdk - app components: Broadcasts. Retrieved from <https://developer.android.com/guide/components/broadcasts.html>
- Górak, R., & Luckner, M. (2016). Modified random forest algorithm for wi-fi indoor localization system. In N. T. Nguyen, L. Iliadis, Y. Manolopoulos, & B. Trawiński (Eds.), *Computational collective intelligence: 8th international conference, iccci 2016, halkidiki, greece, september 28-30, 2016. Proceedings, part ii* (pp. 147-157). Cham: Springer International Publishing.
- Gordon, D., Czerny, J., & Beigl, M. (2014). Activity recognition for creatures of habit. *Personal and Ubiquitous Computing*, 18(1), 205-221. doi:10.1007/s00779-013-0638-2
- Gray, P., & Salber, D. (2001). Modelling and using sensed context information in the design of interactive applications. In Little M.R. & Nigay L. (Eds.), *Engineering for human-computer interaction. Lecture notes in computer science* (Vol. 2254). Berlin, Heidelberg: Springer.
- Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Q.*, 37(2), 337-356. Retrieved
- Gwizdka, J. (2000). What's in the context. *Workshop of the What, Who, Where, When and How of Context Awareness ACM SIGCHI Conference on Human Factors in Computing Systems, 1*. Retrieved from http://comminfo.rutgers.edu/~jacekg/pubs/txt/2000_Context_JGwizdka_W11_Position.pdf
- Harzing, A. W. (2007). Publish or perish. Retrieved from <http://www.harzing.com/pop.htm>
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York, NY, USA: Springer New York Inc.
- Haykin, S. (1998). *Neural networks: A comprehensive foundation*: Prentice Hall PTR.

- Held, A., Buchholz, S., & Schill, A. (2002). Modeling of context information for pervasive computing applications. *Proceedings of SCI 2002/ISAS 2002*. Retrieved from <http://www.rn.inf.tu-dresden.de/uploads/sci2002-paper512jh.pdf>
- Henricksen, K. (2003). *A framework for context-aware pervasive computing application*. (Ph.D), University of Queensland. Retrieved from <http://henricksen.id.au/publications/phd-thesis.pdf>
- Henricksen, K., Indulska, J., & Rakotonirainy, A. (2003). Generating context management infrastructure from high-level context models. *4th International Conference on Mobile Data Management (MDM) - Industrial Track, 1*, 1-6. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.6109>
- Heuer, R. J., & Center for the Study of Intelligence - CIA. (1999). *Psychology of intelligence analysis* (Vol. 1): Center for the Study of Intelligence, Central Intelligence Agency.
- Hevner, A., & Chatterjee, S. (2010). Design science research in information systems. *Integrated Series in Information Systems*, 22, 9-22. doi:10.1007/978-1-4419-5653-8_2
- Hoareau, C., & Satoh, I. (2009). Modeling and processing information for context-aware computing: A survey. *New Generation Computing*, 27(3), 177-196. doi:10.1007/s00354-009-0060-5
- Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., & Retschitzegger, W. (2003). Context-awareness on mobile devices - the hydrogen approach. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, 43, 10 pp. doi:10.1109/hicss.2003.1174831
- Hohpe, G., & Woolf, B. (2004). *Enterprise integration patterns : Designing, building, and deploying messaging solutions*. Boston: Addison-Wesley.
- Hong, D., Schmidtke, H. R., & Woo, W. (2007). Linking context modelling and contextual reasoning. *4th International Workshop on Modeling and Reasoning in Context*.

- Retrieved from
<http://www.academia.edu/download/30694814/10.1.1.83.6279.pdf#page=47>
- HTK. (2016, Jun 01). Htk speech recognition toolkit. <http://htk.eng.cam.ac.uk/>.
- Jackson, P. (1998). *Introduction to expert systems*: Addison-Wesley Longman Publishing Co., Inc.
- Jiming, L., Chi Kuen, W., & Ka Keung, H. (2003). An adaptive user interface based on personalized learning. *IEEE Intelligent Systems*, 18(2), 52-57.
doi:10.1109/mis.2003.1193657
- Julien, C., Payton, J., & Roman, G.-C. (2004). Reasoning about context-awareness in the presence of mobility. *Electronic Notes in Theoretical Computer Science*, 97, 259-276. doi:10.1016/j.entcs.2004.04.040
- Katz, R. H. (1994). Adaptation and mobility in wireless information systems. *IEEE Communications Magazine*, 40(5), 102. doi:10.1109/mcom.2002.1006980
- Kim, J. D., Baik, D.-K., & Son, J. (2012). Ca 5 w 1 h onto: Ontological context-aware model based on 5w1h. *International Journal of Distributed Sensor Networks*. Retrieved from <http://www.hindawi.com/journals/ijdsn/aip/247346/>
- Kindberg, T., & Fox, A. (2004). Ubiquitous system software. *IEEE Pervasive Computing*, 03(03), 57-59. doi:10.1109/mprv.2004.1321029
- Kjærsgaard, M. B. (2007). A taxonomy for radio location fingerprinting. In J. Hightower, B. Schiele, & T. Strang (Eds.), *Location- and context-awareness: Third international symposium, loca 2007, oberpfaffenhofen, germany, september 20-21, 2007. Proceedings* (pp. 139-156). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Kokinov, B. (1999). Dynamics and automaticity of context: A cognitive modeling approach. In Bouquet P., Benerecetti M., Serafini L., Brézillon P., & C. F. (Eds.), *Modeling and using context. Context 1999. Lecture notes in computer science* (Vol. 1688, pp. 200). Berlin, Heidelberg: Springer.

- Korpiää, P., Häkälä, J., Kela, J., Ronkainen, S., & Käsälä, I. (2004). Utilising context ontology in mobile device application personalisation. *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia - MUM '04*, 133-140. doi:10.1145/1052380.1052399
- Kuhn, T. S., & Hacking, I. (2012). *The structure of scientific revolutions* (Fourth edition. ed.). Chicago ; London: The University of Chicago Press.
- Kumar, M., Gupta, A., & Saha, S. (2006). An approach to adaptive user interfaces using interactive media systems. *Proceedings of the 11th international conference on Intelligent user interfaces*, 312. doi:10.1145/1111449.1111521
- Lavie, T., & Meyer, J. (2010). Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, 68(8), 508-524. doi:10.1016/j.ijhcs.2010.01.004
- Lee, K.-C., Kim, J.-H., Lee, J.-H., & Lee, K.-M. (2007). Implementation of ontology based context-awareness framework for ubiquitous environment. *Multimedia and Ubiquitous Engineering*, 278-282. doi:10.1109/mue.2007.136
- Lee, S., Chang, J., & Lee, S. (2011). Survey and trend analysis of context-aware systems. *Information-An International Interdisciplinary Journal*, 14(2). Retrieved from <http://ids.snu.ac.kr/w/images/1/15/Information-liza183.pdf>
- Lee, Y.-S., & Cho, S.-B. (2014). Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data. *Neurocomputing*, 126, 106-115. doi:10.1016/j.neucom.2013.05.044
- Lee, Y. W., Cheatham, T. P., & Wiesner, J. B. (1950). Application of correlation analysis to the detection of periodic signals in noise. *Proceedings of the IRE*, 38(10), 1165-1171. doi:10.1109/JRPROC.1950.233423
- Lenat, D. (1998). *The dimensions of context-space*. Retrieved from <https://courses.csail.mit.edu/6.803/pdf/lenat2.pdf>

- Lenat, D. B., Shepherd, M., Guha, R. V., Pittman, K., & Pratt, D. (1990). Cyc - toward programs with common-sense. *Communications of the ACM*, 33(8), 30-49. doi:10.1145/79173.79176
- Liang, G., & Cao, J. (2015). Social context-aware middleware: A survey. *Pervasive and Mobile Computing*, 17, 207-219. doi:10.1016/j.pmcj.2014.12.003
- Lockhart, J. W., Weiss, G. M., Xue, J. C., Gallagher, S. T., Grosner, A. B., & Pulickal, T. T. (2011). *Design considerations for the wisdom smart phone-based sensor mining architecture*. Paper presented at the Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data, San Diego, California.
- Lukasiewicz, T., & Straccia, U. (2008). Managing uncertainty and vagueness in description logics for the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4), 291-308. doi:10.1016/j.websem.2008.04.001
- Magnusson, C., Waern, A., Gröhn, K. R., Bjernryd, Å., Bernhardsson, H., Jakobsson, A., Salo, J., Wallon, M., & Hedvall, P.-O. (2011). Navigating the world and learning to like it. *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*, 285. doi:10.1145/2037373.2037416
- Mäntyjärvi, J., Himberg, J., Korpipää, P., & Mannila, H. (2001). Extracting the context of a mobile device user. *IFAC Proceedings Volumes*, 34(16), 387-392. doi:10.1016/s1474-6670(17)41555-2
- Manzoor, A., Villalonga, C., Calatroni, A., Truong, H.-L., Roggen, D., Dustdar, S., & Tröster, G. (2010). Identifying important action primitives for high level activity recognition. In P. Lukowicz, K. Kunze, & G. Kortuem (Eds.), *Smart sensing and context: 5th european conference, eurossc 2010, passau, germany, november 14-16, 2010. Proceedings* (pp. 149-162). Berlin, Heidelberg: Springer Berlin Heidelberg.
- McCarthy, J., & Buvac, S. (1997). *Formalizing context (expanded notes)*. Retrieved from Stanford, USA: <http://cogprints.org/00000419>

- Meier, R., & Cahill, V. (2003). Exploiting proximity in event-based middleware for collaborative mobile applications. *Lecture Notes in Computer Science*, 2893, 285-296. doi:10.1007/978-3-540-40010-3_26
- Miluzzo, E., Lane, N. D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S. B., Zheng, X., & Campbell, A. T. (2008). Sensing meets mobile social networks. *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, 337. doi:10.1145/1460412.1460445
- Moore, P., Hu, B., Zhu, X., Campbell, W., & Ratcliffe, M. (2007). A survey of context modeling for pervasive cooperative learning. *2007 First IEEE International Symposium on Information Technologies and Applications in Education*, K5-1-K5-6. doi:10.1109/isitae.2007.4409367
- Mowafi, Y., & Zhang, D. (2007). A user-centered approach to context-awareness in mobile computing. *Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous)*, 1-3. doi:10.1109/mobiq.2007.4450989
- Musolesi, M., Rachuri, K., & Mascolo, C. (2011). Sociablesense: Exploring the trade-offs of adaptive sampling and computation offloading for social sensing. *Proceedings of the 17th annual international conference on Mobile computing and networking*, 73. doi:10.1145/2030613.2030623
- Nadales, M., Röckl, M., & Frank, K. (2010). Comparison of exact static and dynamic bayesian context inference methods for activity recognition. *Pervasive Computing and Communications Workshops*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5470671
- Narayanan, D., Flinn, J., & Satyanarayanan, M. (2000). Using history to improve mobile application adaptation. *Proceedings Third IEEE Workshop on Mobile Computing Systems and Applications*, 31-40. doi:10.1109/mcsa.2000.895379
- Neapolitan, R. E. (1990). *Probabilistic reasoning in expert systems: Theory and algorithms*: John Wiley & Sons, Inc.

- Newman, S. (2015). *Building microservices : Designing fine-grained systems* (First Edition. ed.). Beijing Sebastopol, CA: O'Reilly Media.
- Nicholas, L., Lin, M., Rabi, M., Yang, X., Doryab, A., Lu, H., Ali, S., Choudhury, T., Campbell, A., & Berke, E. (2011). Bewell: A smartphone application to monitor, model and promote wellbeing. *5th Icst/Ieee Conference on Pervasive Computing Technologies for Healthcare*, 23-26. doi:10.4108/icst.pervasivehealth.2011.246161
- Norris, J. R. (1998). *Markov chains* (1st pbk. ed.). Cambridge, UK ; New York: Cambridge University Press.
- O'Connor, N., Cunningham, R., & Cahill, V. (2007). Self-adapting context definition. *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, 336-339. doi:10.1109/saso.2007.48
- Open Data Kit. (2016, Jan 01). Open data kit. <https://opendatakit.org/>.
- OpenSignal. (2015, Aug 01). Opensignal - android fragmentation visualized. <https://opensignal.com/reports/2015/08/android-fragmentation/>.
- Pahl, C., & Lee, B. (2015, 24-26 Aug. 2015). *Containers and clusters for edge cloud architectures -- a technology review*. Paper presented at the 2015 3rd International Conference on Future Internet of Things and Cloud.
- Pascoe, J. (1998). Adding generic contextual capabilities to wearable computers. *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*. doi:10.1109/iswc.1998.729534
- Paymans, T. F., Lindenberg, J., & Neerinx, M. (2004). Usability trade-offs for adaptive user interfaces. *Proceedings of the 9th international conference on Intelligent user interfaces*, 301. doi:10.1145/964442.964512
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414-454. doi:10.1109/surv.2013.042313.00197

- Pl, T., #246, tz, Hammerla, N. Y., & Olivier, P. (2011). *Feature learning for activity recognition in ubiquitous computing*. Paper presented at the Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two, Barcelona, Catalonia, Spain.
- Prado, A. G., & Ortiz, G. (2011). Context-aware services: A survey on current proposals. *Service Computation 2011*, 1(1), 104-109. Retrieved
- Rachuri, K. K., Musolesi, M., Mascolo, C., Rentfrow, P. J., Longworth, C., & Aucinas, A. (2010). Emotionsense. *Proceedings of the 12th ACM international conference on Ubiquitous computing - Ubicomp '10*, 281. doi:10.1145/1864349.1864393
- Raento, M., Oulasvirta, A., Petit, R., & Toivonen, H. (2005). Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4(2), 51-59. doi:10.1109/mprv.2005.29
- Requejo, M. (2007). The role of context in word meaning construction: A case study. *International Journal of English Studies*, 7(1). doi:10.6018/ijes.7.1.48991
- Rialle, V., Lamy, J. B., Noury, N., & Bajolle, L. (2003). Telemonitoring of patients at home: A software agent approach. *Computer Methods and Programs in Biomedicine*. Retrieved from <http://www.sciencedirect.com/science/article/pii/S016926070200161X>
- Roggen, D., Calatroni, A., Förster, K., Tröster, G., Lukowicz, P., Bannach, D., Ferscha, A., Kurz, M., Hözl, G., Sagha, H., Bayati, H., Millán, J. d. R., & Chavarriaga, R. (2011). Activity recognition in opportunistic sensor environments. *Procedia Computer Science*, 7, 173-174. doi:10.1016/j.procs.2011.09.003
- Rosen, R. (2012). Anticipatory systems. *IFSR International Series on Systems Science and Engineering*, 1, 313-370. doi:10.1007/978-1-4614-1269-4_6
- Sanchez, L., Lanza, J., Olsen, R., Bauer, M., & Girod-Genet, M. (2006, 17-21 July 2006). *A generic context management framework for personal networking environments*.

- Paper presented at the 2006 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops.
- Schilit, B., Adams, N., & Want, R. (1994). Context-aware computing applications. *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, 85-90. doi:10.1109/WMCSA.1994.16
- Schilit, B. N., & Theimer, M. M. (1994). Disseminating active map information to mobile hosts. *IEEE Network*, 8(5), 22-32. doi:10.1109/65.313011
- Schmidt, A. (2002). *Ubiquitous computing - computing in context*. (Ph.D), Lancaster University, U.K. . Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.1488&rep=rep1&type=pdf>
- Sheng, Q. Z., & Benatallah, B. (2005, 11-13 July 2005). *Contextuml: A uml-based modeling language for model-driven development of context-aware web services*. Paper presented at the International Conference on Mobile Business (ICMB'05).
- Skeet, J. (2014). *C# in depth* (Third editions. ed.). Shelter Island, NY: Manning.
- Strang, T., & Linnhoff, C. (2004, Jan 01). *A context modeling survey*. Paper presented at the First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004, Nottingham, UK.
- Sun, R., & Giles, C. L. (2001). Sequence learning: From recognition and prediction to sequential decision making. *IEEE Intelligent Systems*, 16(4), 67-70. doi:10.1109/MIS.2001.1463065
- Tangmunarunkit, H., Kang, J., Khalapyan, Z., Ooms, J., Ramanathan, N., Estrin, D., Hsieh, C. K., Longstaff, B., Nolen, S., Jenkins, J., Ketcham, C., Selsky, J., Alquaddoomi, F., & George, D. (2015). Ohmage. *ACM Transactions on Intelligent Systems and Technology*, 6(3), 1-21. doi:10.1145/2717318
- Tuulari, E. (2000). *Context aware hand-held devices*. Retrieved from VTT Technical Research Centre of Finland: <http://www.vtt.fi/inf/pdf/publications/2000/P412.pdf>

- van Sinderen, M. J., van Halteren, A. T., Wegdam, M., Meeuwissen, H. B., & Eertink, E. H. (2006). Supporting context-aware mobile applications: An infrastructure approach. *IEEE Communications Magazine*, 44(9), 96-104. doi:10.1109/mcom.2006.1705985
- Wai Yip, L., & Lau, F. C. M. (2002). A context-aware decision engine for content adaptation. *IEEE Pervasive Computing*, 1(3), 41-49. doi:10.1109/mprv.2002.1037721
- Weiser, M. (1999). The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3), 3-11. doi:10.1145/329124.329126
- World Wide Web Consortium. (2004). *Composite capability/preference profiles (cc/pp): Structure and vocabularies 1.0*.
- Worrall, J., & Currie, G. (1980). *The methodology of scientific research programmes: Volume 1: Philosophical papers*: Cambridge University Press.
- Xu, C., & Cheung, S. C. (2005). Inconsistency detection and resolution for context-aware middleware support. *ACM SIGSOFT Software Engineering Notes*, 30(5), 336-345. doi:10.1145/1095430.1081759
- Yang, J. B., Nguyen, M. N., San, P. P., Li, X. L., & Krishnaswamy, S. (2015). *Deep convolutional neural networks on multichannel time series for human activity recognition*. Paper presented at the Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina.
- Zhang, D., Gu, T., & Wang, X. (2005). Enabling context-aware smart home with semantic web technologies. *International Journal of Human-friendly Welfare Robotic Systems*. doi:10.1.1.469.964
- Zimmermann, A., Lorenz, A., & Oppermann, R. (2007). An operational definition of context. In Kokinov B., Richardson D.C., Roth-Berghofer T.R., & V. L. (Eds.), *Modeling and using context. Context 2007. Lecture notes in computer science* (Vol. 4635, pp. 558-571). Berlin, Heidelberg: Springer.