

Pré-processamento e Tag automático de imagens em ambiente móvel e web, aplicado a um sistema de informação geográfica

Jorge Daniel Marinho dos Santos

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Doutora Maria de Fátima Coutinho Rodrigues

Júri:

Presidente:

Doutor José Antonio Reis Tavares, Professor Adjunto, DEI/ISEP

Vogais:

Doutor João Paulo Jorge Pereira, Professor Adjunto, DEI/ISEP

Doutora Maria de Fátima Coutinho Rodrigues, Professora Coordenadora, DEI/ISEP

Porto, Outubro 2013

*Para a minha família,
Para a minha namorada,
E para aos meus amigos.*

Resumo

No panorama socioeconómico atual, a contenção de despesas e o corte no financiamento de serviços secundários consumidores de recursos conduzem à reformulação de processos e métodos das instituições públicas, que procuram manter a qualidade de vida dos seus cidadãos através de programas que se mostrem mais eficientes e económicos.

O crescimento sustentado das tecnologias móveis, em conjugação com o aparecimento de novos paradigmas de interação pessoa-máquina com recurso a sensores e sistemas conscientes do contexto, criaram oportunidades de negócio na área do desenvolvimento de aplicações com vertente cívica para indivíduos e empresas, sensibilizando-os para a disponibilização de serviços orientados ao cidadão. Estas oportunidades de negócio incitaram a equipa do projeto a desenvolver uma plataforma de notificação de problemas urbanos baseada no seu sistema de informação geográfico para entidades municipais.

O objetivo principal desta investigação foca a idealização, conceção e implementação de uma solução completa de notificação de problemas urbanos de carácter não urgente, distinta da concorrência pela facilidade com que os cidadãos são capazes de reportar situações que condicionam o seu dia-a-dia. Para alcançar esta distinção da restante oferta, foram realizados diversos estudos para determinar características inovadoras a implementar, assim como todas as funcionalidades base expectáveis neste tipo de sistemas. Esses estudos determinaram a implementação de técnicas de demarcação manual das zonas problemáticas e reconhecimento automático do tipo de problema reportado nas imagens, ambas desenvolvidas no âmbito deste projeto. Para a correta implementação dos módulos de demarcação e reconhecimento de imagem, foram feitos levantamentos do estado da arte destas áreas, fundamentando a escolha de métodos e tecnologias a integrar no projeto.

Neste contexto, serão apresentadas em detalhe as várias fases que constituíram o processo de desenvolvimento da plataforma, desde a fase de estudo e comparação de ferramentas, metodologias, e técnicas para cada um dos conceitos abordados, passando pela proposta de um modelo de resolução, até à descrição pormenorizada dos algoritmos implementados.

Por último, é realizada uma avaliação de desempenho ao par algoritmo/classificador desenvolvido, através da definição de métricas que estimam o sucesso ou insucesso do classificador de objetos. A avaliação é feita com base num conjunto de imagens de teste, recolhidas manualmente em plataformas públicas de notificação de problemas, confrontando os resultados obtidos pelo algoritmo com os resultados esperados.

Palavras-chave: notificação de problemas urbanos, classificação automática, anotações em imagem, reconhecimento de imagem, sistemas de informação geográficos, aplicações móveis

Abstract

In the present socio-economic scenario, cost containment and fund cutting for secondary services that consume resources, lead to the redesign of processes and methods of public institutions, while seeking to maintain the quality of life of citizens through programs that prove to be more efficient and economic.

The sustained growth of mobile technologies, in conjunction with the appearance of new person-machine interaction paradigms making use of sensors and context aware systems, created new business opportunities in the area of the application development with civic concerns for individuals and businesses, sensitizing them for the creation of services targeted to citizens. These business opportunities encouraged the project team to develop a platform for the notification of urban problems based on their geographic information system for municipal authorities.

The main objective of this research focuses on the idealization, design and implementation of a complete solution for the notification of urban problems of non-urgent nature, distinct from the competition by the ease with which citizens are able to report situations that affect their daily lives. To achieve this distinction of the remaining offers on the market, several studies were conducted to determine which innovative features to implement, as well as all the basic features expected in this kind of systems. These studies determined the implementation of techniques for the manual demarcation of the problem boundaries within an image as well as the automatic recognition of its type all based on the attached pictures of reports, both developed in this project. For the proper development of the demarcation and image recognition modules, studies were made about the state of art of applications in each one of these areas, supporting the selection of methods and technologies to integrate in the project.

In this context, the various phases that make the development process of the platform will be presented in further detail, from the initial study and comparison of tools, methodologies, and techniques for each of the concepts addressed, through the proposal of a resolution model until the in depth description of the implemented algorithms.

Finally, a performance evaluation of the developed algorithm/classifier pair is conducted through the definition of metrics that estimate the success or failure of the objects classifier. The assessment is based on a set of test images, collected manually on public problem notification platforms, comparing the results obtained by the algorithm with the expected results.

Keywords: notification of urban problems, automatic classification, image annotations, image recognition, geographic information systems, mobile applications

Agradecimentos

Em primeiro lugar, gostaria de agradecer à minha orientadora, Professora Doutora Fátima Rodrigues, pelos conhecimentos transmitidos e apoio científico prestado, assim como todo o empenho, dedicação, disponibilidade e paciência demonstrados. A sua participação ativa tornou esta investigação uma experiência verdadeiramente enriquecedora a nível académico e profissional.

Gostaria igualmente de agradecer ao meu orientador técnico, Engenheiro Lino Oliveira, pela idealização e forte aposta neste projeto, e também por todo o apoio, sugestões de melhoramento e correções. O seu acompanhamento e contribuição diários fizeram deste projeto uma excelente oportunidade para aquisição de competências e crescimento pessoal e profissional.

Ao Instituto de Engenharia de Sistemas e Computadores do Porto (INESC Porto), em particular à Unidade de Sistemas de Informação e Computação Gráfica, pela oportunidade concedida e meios facultados para a realização de toda esta investigação.

A todos os meus colegas do INESC Porto que partilham diariamente o seu conhecimento e experiência, e estão dispostos a ajudar em todos os momentos. Em especial, gostaria de agradecer aos elementos da minha equipa de projeto, Rui Barros, Leonel Dias e André Rodrigues, assim como ao Rúben, Lília, Marta, Paulo Melo, José Carlos, Ricardo Henriques, Carlos Almeida, Carlos Aldeias, Artur Capela e Lucian Ciobanu, pelo incentivo e companheirismo constantes.

A todos os docentes do Instituto Superior de Engenharia do Porto (ISEP) que me acompanharam ao longo do meu percurso académico, e possibilitaram a contínua aquisição e evolução de conhecimentos.

Aos meus amigos, em particular ao Rui Araújo, Ricardo Ramalho, César Soares, Leandro Loureiro, Tiago Seabra, João Silva, Sofia Fernandes, Bruno Santos, Bárbara Santos, Rúben Ausina e Joana Machado, por todos os momentos de descontração, assim como todo o ânimo e incentivo que me conferiram.

À minha família, em especial aos meus pais, irmãos, cunhadas e sobrinhas, pelo apoio, fraternidade e compreensão totais nos momentos em que estive mais ausente.

E a ti Andreia, por todo o amor, afeto, e motivação para continuar quando parecia tão mais fácil desistir. Provaste mais uma vez ser um dos pilares da minha vida, e sem o teu apoio incondicional tudo seria mais difícil.

A todos os que referi, e aos que possa não ter referido, o meu mais sincero Obrigado!

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação	2
1.3	Principais objetivos e descrição sumária do trabalho efetuado	3
1.4	Estrutura do Documento	4
2	Notificação de problemas urbanos	7
2.1	Introdução	7
2.2	Sistemas de notificação de problemas urbanos	9
2.2.1	FixMyCity	9
2.2.2	FixMyStreet	10
2.2.3	CitySourced	11
2.2.4	SeeClickFix	12
2.2.5	A Minha Rua (eCivitas)	13
2.2.6	CivicTicket	14
2.2.7	Snap Send Solve	15
2.2.8	PDX Reporter	16
2.3	Análise comparativa das soluções de notificação de problemas urbanos identificadas	17
2.4	Conclusões	22
3	Anotações em imagem	25
3.1	Introdução	25
3.2	Aplicações existentes	27
3.3	Tecnologias de anotação em imagem	29
3.3.1	LEADTOOLS - Document Image Annotations	29
3.3.2	Android Graphics 2D API	30
3.4	Análise comparativa das tecnologias de anotação estudadas	31
3.5	Conclusões	32
4	Reconhecimento de padrões em imagem	33
4.1	Introdução	33
4.2	Aplicações existentes	35
4.2.1	Google Goggles	36
4.2.2	Visidon AppLock	38
4.3	Tecnologias de reconhecimento em dispositivos móveis	40
4.3.1	OpenCV	40
4.3.2	FastCV	41
4.3.3	Camellia Library	41

4.3.4	Neuroph	42
4.3.5	Moodstocks	43
4.3.6	Kooaba	44
4.4	Análise comparativa das tecnologias de reconhecimento analisadas	45
4.5	Conclusões	46
5	Implementação e Avaliação	47
5.1	Introdução	47
5.2	Sistema de informação geográfico	49
5.2.1	Portal SIG	49
5.2.2	Aplicações Setoriais.....	51
5.3	Modelo proposto	53
5.3.1	Requisitos funcionais e não funcionais	54
5.3.2	Arquitetura da solução	60
5.4	Implementação	63
5.4.1	Tecnologias utilizadas	64
5.4.2	Base de Dados.....	65
5.4.3	Camada de serviços	68
5.4.4	Aplicação móvel	70
5.4.5	Módulo de comunicação e serviços.....	74
5.4.6	Módulo de anotações em imagem	75
5.4.7	Módulo de reconhecimento de imagem	78
5.5	Caso de estudo.....	83
5.5.1	Definição do Caso de Estudo	84
5.5.2	Análise do Caso de Estudo	86
5.5.3	Conclusões.....	91
6	Conclusões	93
6.1	Resumo	93
6.2	Objetivos Alcançados	95
6.3	Limitações e Trabalho Futuro	96
6.4	Considerações Finais	97

Lista de Figuras

Figura 1 – Skitch - Adição de uma anotação numa imagem	27
Figura 2 – Editor de imagem – Edição de uma fotografia [Google, 2013b]	28
Figura 3 – Adobe Photoshop Touch – Manipulação de uma fotografia	28
Figura 4 – LEADTOOLS - Anotações em documentos	29
Figura 5 – Drawing API – Demonstração de potencialidades.....	30
Figura 6 – Google Goggles – Reconhecimento de caracteres	36
Figura 7 – Google Goggles – Reconhecimento de padrões e classificação	37
Figura 8 – AppLock – Reconhecimento de faces	38
Figura 9 – GeoAve – Arquitetura do portal	49
Figura 10 – GeoAve – Mapa	50
Figura 11 – GeoAve – Administração do Mapa	51
Figura 12 – GeoAve – CleanMyCity.....	52
Figura 13 – Diagrama de casos de uso – Sistema (nível zero)	54
Figura 14 – Diagrama de casos de uso – Aplicação móvel	55
Figura 15 – Arquitetura da solução proposta [Santos et al., 2013]	60
Figura 16 – Módulos constituintes da aplicação móvel	61
Figura 17 – Diagrama de Entidade-Relacionamento (Notação <i>Crow’s Foot</i>)	66
Figura 18 – Algoritmo de determinação dos relatórios de ocorrências a apresentar	68
Figura 19 – Aplicação móvel – Listagem de ocorrências.....	73
Figura 20 – Módulo de serviços e comunicação – Diagrama de classes.....	74
Figura 21 – Módulo de anotações em imagem – Diagrama de classes	75
Figura 22 – Módulo de anotações em imagem – Algoritmo para desenho de linhas	76
Figura 23 – Aplicação móvel – Demarcação manual de patologia na estrada	77
Figura 24 – Módulo de reconhecimento de imagem – Diagrama de classes.....	79
Figura 25 – Algoritmo de deteção e classificação de objetos	80
Figura 26 – Aplicação móvel – Reconhecimento de objetos na imagem.....	82
Figura 27 – Aplicação móvel – Atribuição de uma categoria (classificação)	82
Figura 28 – Gráfico de comparação dos resultados do reconhecimento	87
Figura 29 – Gráfico de comparação do número de falsos positivos e falsos negativos.....	90

Lista de Tabelas

Tabela 1 — Notificação de problemas urbanos: Comparação dos projetos analisados.....	18
Tabela 2 — Tecnologias de anotação - Comparação das soluções encontradas	31
Tabela 3 — Tecnologias de reconhecimento - Comparação das soluções encontradas	45
Tabela 4 — Subdivisão das imagens recolhidas	84
Tabela 5 — Parâmetros de configuração dos utilitários de criação do classificador	85
Tabela 6 — Resultados do reconhecimento das imagens de teste	86
Tabela 7 — Resultados coincidentes segundo o coeficiente de correlação de Pearson	88
Tabela 8 — Falsos Positivos e Falsos Negativos	89

Acrónimos e Símbolos

Lista de Acrónimos

API	<i>Application Programming Interface</i>
ADN	Ácido Desoxirribonucleico
BSD	<i>Berkeley Software Distribution</i>
DICOM	<i>Digital Imaging and Communications in Medicine</i>
GPL	GNU General Public License
GPS	<i>Global Positioning System</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
KML	<i>Keyhole Markup Language</i>
LBP	<i>Local Binary Patterns</i>
OCR	<i>Optical Character Recognition</i>
OpenCV	<i>Open Source Computer Vision</i>
RLE	<i>Run-Length Encoded</i>
SDK	<i>Software Development Kit</i>
SGBD	Sistema Gestor de Bases de Dados
SIG	Sistema de Informação Geográfico
SOAP	<i>Simple Object Access Protocol</i>
WFS	<i>Web Feature Service</i>
WMS	<i>Web Map Service</i>
WSDL	<i>Web Services Description Language</i>
XML	<i>eXtensible Markup Language</i>

1 Introdução

1.1 Enquadramento

No panorama socioeconómico atual, a contenção de despesas e o corte no financiamento de serviços secundários consumidores de recursos conduzem à reformulação de processos e métodos das instituições públicas, que procuram manter a qualidade de vida dos seus cidadãos através de programas que se mostrem mais eficientes e económicos. O crescimento sustentado das tecnologias móveis, em conjunção com o aparecimento de novos paradigmas de interação pessoa-máquina com recurso a sensores e sistemas conscientes do contexto, criaram oportunidades de negócio na área do desenvolvimento de aplicações com vertente cívica para indivíduos e empresas, sensibilizando-os para a disponibilização de serviços orientados ao cidadão [Tamilin et al., 2012].

A notificação de problemas urbanos é uma das áreas que já sofreu alterações na sua forma de interagir com os cidadãos, necessitando de se adaptar à realidade das grandes cidades urbanas. Os sistemas de notificação clássicos, como linhas telefónicas e divulgação de endereços de correio eletrónico, obrigavam as entidades municipais à contratualização de operadores de centros de chamadas telefónicas (*callcenters*) para garantir a correta receção e triagem das queixas dos cidadãos, o que comportava custos elevados às instituições. Além disso, a introdução de mediadores na comunicação entre os cidadãos e o poder local pode originar falhas de interpretação dos conteúdos reportados e conduzir à classificação deficiente do problema.

A evolução destes sistemas de resolução de problemas passará, provavelmente, pela utilização das tecnologias móveis, reaproveitando a presença deste tipo de dispositivos no dia-a-dia dos utilizadores, facilitando o processo de notificação junto das entidades competentes. Esta interface, natural para os cidadãos, minimiza o esforço necessário para contribuir positivamente para o desenvolvimento da comunidade local, tornando os cidadãos elementos com uma participação ativa na sociedade em que se inserem. Problemas como carros abandonados na via pública, bocas-de-incêndio danificadas, buracos na estrada, edifícios grafitados, mobiliário urbano danificado ou objetos a bloquear a via, apesar de

carecerem de uma resolução não urgente, podem afetar gravemente a qualidade de vida e o bem-estar dos cidadãos, e interferir com o decorrer da sua vida quotidiana.

A equipa do projeto agarrou esta oportunidade de negócio e idealizou uma solução para a notificação de problemas baseada no seu sistema de informação geográfico para entidades municipais. Este sistema conta já com uma aplicação móvel direcionada para os cidadãos, responsável pelo setor da recolha de lixo dos municípios. Os conhecimentos detidos pela equipa neste domínio de aplicações, em conjugação com a necessidade de uma aplicação setorial para o ato de exercer e promover atos de cidadania, reuniram as condições ideais para a conceção e desenvolvimento de uma plataforma para gestão de notificações de problemas urbanos orientada aos cidadãos.

O objetivo principal desta investigação passa pelo desenvolvimento de uma aplicação para permitir a submissão de relatórios de problemas urbanos por parte de residentes de uma determinada localidade, encaminhando automaticamente o processo para as entidades competentes responsáveis pela sua resolução. Para diferenciar este produto da concorrência, foram estudadas várias ferramentas do mercado que oferecem serviços nesta área de negócio, e identificadas as lacunas comuns a todos estes sistemas. Para além desta análise de mercado, foram também estudados métodos para a demarcação manual de zonas problemáticas em fotografias, assim como a classificação das imagens segundo tipos de problemas bem definidos. Estes métodos visam, por um lado, reduzir o tempo de preenchimento dos relatórios de ocorrências pelos cidadãos, na medida em que podem anexar uma fotografia da ocorrência e facilmente realçar a zona problemática, não necessitando de descrições textuais extensas. Estas técnicas pretendem ainda eliminar a ambiguidade existente no processamento dos relatórios de ocorrências por parte das entidades responsáveis, através da atribuição automática de uma categorização do problema reportado recorrendo a processos de reconhecimento da imagem.

1.2 Motivação

A revolução ocorrida no mundo tecnológico nos últimos anos, muito devido à massificação da computação móvel, incitou o aparecimento de aplicações para auxiliar nas tarefas do dia-a-dia dos utilizadores, através do uso de dispositivos móveis como *smartphones* ou *tablets*. A usabilidade, acessibilidade, satisfação e sensação de conforto que estes dispositivos conferem aos seus utilizadores, criam experiências verdadeiramente imersivas, tornando a computação ubíqua uma realidade evidente através do recurso a sensores precisos e conscientes do contexto. Após o aparecimento destes dispositivos, foram geradas inúmeras áreas de negócio capazes de explorar o seu potencial, como o desenvolvimento de aplicações de foro cívico e serviços orientados ao cidadão.

Este tipo de aplicações orientadas ao dever cívico, constituem uma excelente oportunidade de negócio, uma vez que permitem que as entidades municipais diminuam os seus custos de operação, ao mesmo tempo que os cidadãos assumem um papel ativo na sua comunidade

local, promovendo o crescimento e prosperidade da sociedade em que estão inseridos. O retorno de investimento do mercado global das aplicações móveis atrai a investigação e desenvolvimento tecnológico nesta área, pelo que já existem soluções para a notificação de problemas urbanos implantadas no mercado. A existência de processos de padronização da estrutura de relatórios de ocorrências, assim como a disponibilização de serviços universais de consulta e notificação de problemas, geram concorrência neste ramo, levando ao aparecimento de alternativas aos sistemas atuais e à reinvenção de funcionalidades. No entanto, a adaptação deste tipo de sistemas às constantes alterações da realidade tecnológica é feita a um baixo ritmo, deixando margem para o aparecimento de soluções inovadoras.

Estas oportunidades despertaram o interesse da equipa do projeto, já especialista no domínio de sistemas e aplicações georreferenciados para entidades municipais. A recente implementação de uma aplicação do foro cívico para promover a eficiência na recolha de resíduos alavancou o processo de conceptualização da plataforma para a notificação de problemas urbanos e lançou o mote para a discussão dos sistemas atuais. Reaproveitando toda a infraestrutura de base do portal SIG, e seguindo a lógica de reutilização das suas camadas, a solução a implementar passa pelo desenvolvimento de uma aplicação setorial e serviços associados. A experiência detida pela equipa do projeto neste nicho de aplicações permite comprovar a dependência de utilizadores ativos na alimentação da plataforma, e assim atingir o sucesso do projeto. Nesta linha de pensamento, foram reutilizados e aplicados conceitos de *crowd-sourcing*¹, com o propósito de repartir o volume de informação a processar pelos cidadãos participativos na comunidade.

Considerando as motivações supramencionadas, o propósito desta investigação reside na construção de uma plataforma de notificação de problemas urbanos não urgentes que se diferencie dos atuais através de uma nova lógica de preenchimento dos relatórios. A inovação neste campo será atingida pela inclusão de um módulo de anotação das fotografias anexadas e de um módulo de reconhecimento automático das patologias urbanas mais comuns, eliminando a necessidade da descrição pormenorizada das ocorrências.

1.3 Principais objetivos e descrição sumária do trabalho efetuado

O principal objetivo desta investigação, prende-se com o desenvolvimento de uma plataforma para a notificação de problemas urbanos, que se diferencie da concorrência pela facilidade com que os cidadãos reportam situações que condicionem o seu dia-a-dia. Do ponto de vista das entidades que executam os processos de triagem, encaminhamento e resolução das ocorrências, as descrições textuais dos problemas podem originar erros de interpretação, sendo necessário assegurar mecanismos adicionais que clarifiquem a mensagem a transmitir.

¹ *Crowd-sourcing* define-se como o ato de delegar uma tarefa normalmente executada por um determinado agente (pessoa ou entidade) a um grande grupo de pessoas não organizado ou definido [Howe, 2006].

Assim sendo, esta investigação pretende também estudar as boas práticas implementadas noutros sistemas, e integrar métodos para a demarcação manual das zonas problemáticas em imagens, e o reconhecimento automático do tipo de problema reportado com base nas fotografias submetidas.

Para poder construir uma aplicação que traga valor acrescentado aos cidadãos, foi necessário efetuar um levantamento do estado da arte da área de notificação de problemas urbanos, para determinar, por um lado, as funcionalidades base que os utilizadores esperam deste tipo de aplicações, e por outro lado identificar lacunas comuns a todos os sistemas.

As conclusões retiradas deste estudo permitiram identificar funcionalidades nucleares inexistentes nos sistemas típicos de notificação de problemas, ambas para facilitar o ato de reportar: a demarcação manual de zonas problemáticas em imagens, e o reconhecimento de tipos de problemas com base nas fotografias submetidas. Para a integração destas funcionalidades no sistema, foi também realizado um levantamento do estado da arte para os módulos de anotação e reconhecimento de objetos em imagens.

Após o término destes levantamentos e estudos comparativos de soluções, foi desenvolvida a aplicação móvel e os seus módulos constituintes, a par da implementação da camada de serviços *Web*.

Finda a fase de desenvolvimento, os algoritmos para reconhecimento de imagens, como característica mais inovadora do sistema, são devidamente testados com dados que simulam situações reais, sendo por fim efetuada uma análise e interpretação dos resultados obtidos.

1.4 Estrutura do Documento

Nesta secção é apresentada a estrutura deste documento, fornecendo uma visão geral do tema a ser desenvolvido. A organização deste documento rege-se pela divisão lógica das matérias abordadas em seis capítulos, cujos assuntos são sucintamente descritos de seguida:

- **Capítulo 1:** Este capítulo trata de enquadrar o tema do trabalho desenvolvido, apresentar os fatores de motivação que conduziram à sua concretização, expor os objetivos a atingir e resumir brevemente as atividades realizadas;
- **Capítulo 2:** Detalha o levantamento do estado da arte das aplicações de notificação de problemas urbanos, apresenta o estudo comparativo elaborado e as conclusões decorrentes da análise desse estudo;
- **Capítulo 3:** Resume o estudo efetuado sobre aplicações e tecnologias móveis de anotação em imagem;
- **Capítulo 4:** Apresenta o resultado do levantamento do estado da arte de aplicações e tecnologias móveis com capacidades de reconhecimento de padrões em imagem;

- **Capítulo 5:** Tem por objetivo descrever pormenorizadamente as várias fases de desenvolvimento do projeto, recorrendo a esquemas, excertos de algoritmos, e imagens para ilustrar o processo. É ainda apresentado um caso de estudo para avaliação e teste dos componentes desenvolvidos;
- **Capítulo 6:** Neste capítulo são retiradas conclusões sobre o trabalho realizado, apresentadas as suas limitações, e identificadas linhas de pensamento para desenvolvimentos futuros.

2 Notificação de problemas urbanos

2.1 Introdução

O conceito inerente à notificação de problemas urbanos está intimamente relacionado com a necessidade de participação ativa do cidadão na comunidade em que se insere, contribuindo não só para o seu crescimento enquanto indivíduo, como também para o desenvolvimento da sociedade civil da sua localidade. A qualidade de vida e o bem-estar dos cidadãos pode ser afetado gravemente por problemas que perturbem o normal decorrer da vida quotidiana, como carros abandonados na via pública, bocas-de-incêndio danificadas, buracos na estrada, edifícios grafitados, mobiliário urbano danificado, objetos a bloquear a via, entre outros. Apesar de existirem meios alternativos de comunicação com as entidades responsáveis (por exemplo, a linha telefónica 311 na cidade de Nova Iorque para situações não urgentes), estas opções nunca ganharam a devida notoriedade, muito devido à natureza não urgente dos problemas [Foth et al., 2011].

No contexto mundial atual, a contenção de despesas e o corte de serviços que consomem fundos públicos são a palavra de ordem no que diz respeito às medidas adotadas para aumentar o crescimento económico dos países [Lusa, 2013]. Isto leva a que os governos locais tenham de fazer mais com cada vez menos, forçando os seus departamentos a reinventarem os seus métodos e procedimentos e simultaneamente procurarem manter a qualidade de vida dos seus cidadãos, através de programas que se mostrem mais eficientes e económicos. O aparecimento de novos paradigmas de interação entre as pessoas e os sistemas, nomeadamente a interação com sistemas móveis baseada em sensores e sistemas conscientes do contexto, despertou a atenção das empresas para o desenvolvimento de aplicações com vertente cívica, e sensibilizou-as para a disponibilização de serviços orientados ao cidadão [Tamilin et al., 2012]. Para as empresas, o retorno do investimento neste segmento de aplicações é justificado pela franca expansão do mercado dos dispositivos móveis (segundo [Reed, 2013], o número de dispositivos Android ativados até Março de 2013 ronda os 750 milhões de dispositivos) e este crescimento exponencial serve como alavanca para este tipo de sistemas, uma vez que o público-alvo está em constante evolução

tecnológica. Os sistemas orientados ao dever cívico já não são novidade [Foth et al., 2011]. No entanto, devido à massificação do uso de dispositivos móveis, a forma como os cidadãos interagem com as entidades de gestão governamental local sofreu grandes alterações.

Considerando as oportunidades de negócio supramencionadas, aliadas à necessidade da integração de aplicações setoriais num sistema de informação geográfico existente, reuniram-se as condições necessárias para o desenvolvimento de uma aplicação orientada ao cidadão e ao ato de exercer a cidadania. A aplicação a desenvolver permitirá aos cidadãos de uma determinada localidade notificarem as entidades institucionais competentes de problemas não urgentes que ocorram na sua localidade, servindo como o canal de comunicação primordial entre os cidadãos e os órgãos de gestão local para resolução deste tipo de problemas. Seguindo os moldes de aplicações existentes no mercado, e as diretivas resultantes de uma normalização e estruturação destas aplicações (especificação Open311²), a solução a implementar possibilitará a criação de relatórios de ocorrências de problemas locais como vias públicas danificadas, mobiliário urbano danificado, grafitis em locais não autorizados, lixo na via pública, objetos a bloquear acessos, entre outros.

Apesar de se mostrar uma área com potencial para expansão, existem atualmente sistemas que dão uma resposta concreta a este tipo de necessidades da comunidade, tanto no panorama nacional como internacional. Estas soluções assemelham-se na oferta que disponibilizam aos utilizadores, notando-se apenas algumas diferenças em pequenas funcionalidades e na abordagem tomada pelas empresas para a comercialização do produto. Embora a oferta atual se baseie em moldes pré-estabelecidos para colmatar as necessidades específicas dos cidadãos, é importante estudar as várias alternativas existentes para, simultaneamente, definir o conjunto de funcionalidades comuns e imprescindíveis para os utilizadores que já se constituem como dado adquirido e encontrar os seus pontos divergentes para que seja possível delinear uma proposta com valor para os utilizadores, aglomerando o que de melhor existe atualmente com a inovação pretendida neste projeto. As soluções apresentadas neste capítulo representam o estado da arte dos sistemas de *civic reporting*, disponibilizando um leque de funcionalidades e serviços para cidadãos, entidades municipais e a imprensa.

Este capítulo procura enquadrar a temática de notificação de problemas urbanos no contexto do projeto a desenvolver, apresentando sucintamente cada uma das principais soluções para a notificação de problemas urbanos disponíveis no mercado, comparando estes projetos mediante vários critérios para clarificar os seus pontos comuns e divergentes, e também identificar as lacunas de todos estes sistemas. Por fim, são extraídas conclusões do estudo efetuado e feita uma retrospectiva dos sistemas analisados, retirando o contributo positivo que conferem à comunidade.

² Iniciativa Open311 – Especificação para sistemas que notificam problemas não urgentes em espaços públicos [Open311, 2013a]. Mais detalhe em <http://open311.org/>

2.2 Sistemas de notificação de problemas urbanos

2.2.1 FixMyCity

O projeto FixMyCity constitui uma plataforma modular para a criação de sistemas de *Civic Reporting*, desenvolvida pelo Instituto Fraunhofer FOKUS. Este tipo de plataforma possibilita aos cidadãos reportarem problemas que encontrem na via pública (maioritariamente na sua área de residência), para notificar e informar os vários órgãos de gestão governamental da localidade dessas mesmas situações. Neste projeto, a combinação dos vários componentes que constituem a solução FixMyCity permitem o estabelecimento de canais bidirecionais entre os cidadãos e a administração pública local, sendo possível o acompanhamento e o envolvimento de ambas as entidades no evoluir de um dado problema, desde a sua deteção até à sua resolução [Fraunhofer FOKUS, 2013]. O projeto tem como objetivo chegar ao maior número de cidadãos possível, estabelecendo como prioridades o suporte multiplataforma a uma grande variedade de dispositivos e sistemas operativos, aliado com uma forte integração com as redes sociais [Fraunhofer FOKUS, 2013].

Das várias potencialidades do sistema, destacam-se:

- Partilha de ocorrências nas várias redes sociais;
- Processo de submissão de ocorrências simplificado (preenchimento minimalista dos detalhes de uma ocorrência);
- Arquitetura modular/distribuída (separação clara da API das várias aplicações e módulos existentes);
- Contexto georreferenciado (pesquisa de ocorrências numa determinada zona de abrangência);
- Categorização de ocorrências;
- Geração de alertas para os *media* locais.

A arquitetura adotada neste sistema promove o baixo acoplamento dos diversos componentes envolvidos, através da utilização de uma API que disponibiliza todas as operações passíveis de serem realizadas no sistema. Todas as aplicações e módulos executam ações na plataforma através da invocação de métodos da API, o que possibilita o desenvolvimento de aplicações por parte de terceiros para consumir também estes serviços.

No momento da escrita deste documento, a solução ainda não se encontra disponível para o grande público, estando ainda na fase inicial da sua divulgação enquanto produto [Fraunhofer FOKUS, 2013].

2.2.2 FixMyStreet

FixMyStreet é uma solução para possibilitar aos cidadãos do Reino Unido visualizar, reportar e discutir problemas que encontrem na sua localidade. É uma solução modular, e constituída por vários componentes: uma plataforma *Web*, aplicações móveis nos principais sistemas operativos para os cidadãos reportarem situações, e uma plataforma de *Backoffice* para os governos locais darem seguimento às ocorrências. Constitui uma ferramenta de *Citizen Reporting* especializada na área dos problemas existentes na via pública, suportando um conjunto fechado de objetos suscetíveis de serem reportados: veículos abandonados, grafitis, lixo (vidros na estrada), postes de iluminação danificados, buracos na estrada, entre outros.

Desde o seu lançamento em 2007, já se expandiu para outros países e inspirou também o desenvolvimento de projetos similares como o FixMyStreet.ca (Canada), Mark-a-Spot (Alemanha), FixmyStreet.kr (Coreia) [Steinberg, 2011]. Este projeto foi desenvolvido pela mySociety, uma organização sem fins lucrativos que se dedica ao desenvolvimento de projetos com impacto social [mySociety, 2013].

Este projeto foi pioneiro no teste às potencialidades do *crowd-sourcing* [Foth et al., 2011], permitindo o envio de notificações pela *Web* de forma anónima. A disponibilização pública de todas as ocorrências reportadas previne a duplicação de notificações sobre o mesmo problema, ao mesmo tempo que fomenta a discussão *online* dessas mesmas ocorrências. A discussão gerada e a troca de ideias por parte dos cidadãos demonstra um sentido cívico apurado por parte das pessoas da comunidade local, e que muitas vezes resulta na descoberta e adoção de soluções dadas pelos próprios cidadãos.

As características que se evidenciam no projeto FixMyStreet são:

- Tratamento estatístico das ocorrências por localidade;
- Pesquisa de notificações por localidade;
- Subscrição de alertas para notificações que ocorram numa área específica (o utilizador é livre de escolher uma localidade ou indicar um raio a partir de uma rua concreta);
- Submissão de fotografias na notificação do problema;
- Discussão dos utilizadores sobre ocorrências.

Assente nos moldes de um verdadeiro projeto *open-source* de e para a comunidade, todo o código fonte está acessível no repositório Git disponível no popular serviço de alojamento de código GitHub (<https://github.com/mysociety/fixmystreet>), dando a possibilidade de qualquer cidadão poder ajudar no desenvolvimento do projeto.

2.2.3 CitySourced

O projeto CitySourced é uma plataforma que promove a participação e o envolvimento dos cidadãos dos Estados Unidos da América na gestão de ocorrências não prioritárias (não urgentes) em tempo real. Define-se como uma plataforma simples e intuitiva, possibilitando aos residentes de uma determinada localidade reportarem problemas (situações que coloquem em causa a segurança pública, qualidade de vida, questões ambientais, entre outros) às entidades responsáveis pertencentes ao governo local para futura resolução. O CitySourced fornece também aos governos de gestão local a oportunidade de utilizarem o sistema para obter *feedback* dos seus residentes, promovendo uma interação rica, positiva e colaborativa, permitindo poupar tempo e dinheiro na deteção de problemas de cariz cívico [CitySourced, 2009].

Das várias funcionalidades existentes nesta solução, ganham especial relevância:

- A forte integração com as aplicações existentes nos sistemas de gestão municipal (compatibilidade com grande parte dos sistemas utilizados na gestão autárquica);
- Diversidade na forma como se notificam problemas: é possível anexar qualquer tipo multimédia (fotografias, áudio, vídeo);
- Todas as ocorrências reportadas têm intrinsecamente associada informação georreferenciada, inclusive a indicação da direção na qual foi tirada a fotografia da ocorrência;
- Possibilidade de notificação de problemas no anonimato (apesar de ser necessária a confirmação de um endereço de e-mail, os detalhes do utilizador não são divulgados).

Este sistema distribuído sobre uma licença própria de código fechado, disponibiliza a solução e presta serviços sujeitos a pagamento e apenas aos municípios dos Estados Unidos da América [CitySourced, 2009].

A empresa dedicada ao desenvolvimento constante desta plataforma (denominada igualmente de CitySourced), para se diferenciar da concorrência presta serviços especializados de instalação e suporte aos municípios para minimizar o esforço destes na migração ou reutilização dos sistemas atuais de gestão de ocorrências [CitySourced, 2009].

O projeto CitySourced conta já com aplicações desenvolvidas para os maiores sistemas operativos móveis existentes atualmente: Android, iPhone, Windows Phone e Blackberry, tendo já em vista o desenvolvimento para PalmOS e Symbian. A compatibilidade com todos estes sistemas operativos permite que esta solução alcance o maior número de utilizadores possível [CitySourced, 2009].

2.2.4 SeeClickFix

O projeto SeeClickFix possibilita aos cidadãos de todo o Mundo reportarem diversos problemas da sua área de residência como veículos abandonados, mobiliário urbano danificado, paredes grafitadas entre outros, constituindo uma solução completa para notificação, seguimento e resolução destes problemas. Este projeto segue uma arquitetura modular e é constituído por vários componentes: uma plataforma *Web*, aplicações móveis, e uma API que permite o acesso a toda a informação. Relativamente à componente móvel, o SeeClickFix conta já com aplicações desenvolvidas para os principais sistemas operativos móveis: iPhone, Android e Blackberry. Dispõe atualmente de uma aplicação SeeClickFix genérica nas lojas de aplicações destes S.O. móveis (até à data de utilização gratuita), mas a empresa trata também de criar aplicações personalizadas para os seus clientes, permitindo alterar o logotipo, nome, descrição, etc. (por exemplo, DC 311) [SeeClickFix, 2013].

A solução adota uma perspetiva com foco no cidadão, através da recompensa em pontos cívicos aos utilizadores mais ativos, seja a reportar, comentar, divulgar ou votar ocorrências, o que envolve o utilizador em todo o processo de tratamento das situações.

De entre os vários pontos fortes da solução, demarcam-se:

- A possibilidade de configurar os tipos de ocorrências a reportar por área (geralmente por município), assim como as questões a colocar ao utilizador aquando da notificação de ocorrências de um determinado tipo;
- Partilha das ocorrências nas principais redes sociais;
- Foco no cidadão (partilha nas redes sociais e recompensas em pontos cívicos aos cidadãos mais ativos);
- Promove a interação entre vizinhos e a discussão dos problemas locais;
- Sistema de priorização (ocorrências mais votadas ou discutidas pelos utilizadores ganham especial relevância);
- Definição de áreas de controlo (que os cidadãos poderão subscrever para receber notificações de novas ocorrências) através da demarcação livre de limites no mapa.

Com uma forte ideologia sobre a necessidade do envolvimento e pro-atividade dos cidadãos no que diz respeito ao papel que podem desempenhar em prol da comunidade local onde residem, os criadores do projeto SeeClickFix (originários de New Haven, Connecticut, EUA) procuram tornar o mundo um lugar melhor, dando voz aos cidadãos para apontarem situações problemáticas na sua zona e interagirem com as entidades do governo local responsáveis pela resolução destas, aproximando assim a comunidade das entidades municipais [SeeClickFix, 2013].

2.2.5 A Minha Rua (eCivitas)

O projeto A Minha Rua fez parte do Programa Simplex 2009 desenvolvido pela Agência para a Modernização Administrativa, com vista à remodelação e simplificação de processos relacionados com a Administração Pública de Portugal. Atuando em conjunto com outras medidas governamentais como a Empresa na Hora ou a Empresa Online [Agência para a Modernização Administrativa, 2009b], este projeto pretende chegar a todos os cidadãos de Portugal através de uma interface *Web* simplista seguindo o lema do programa: “Quanto mais simples, melhor”, permitindo a notificação junto das entidades camarárias e instituições responsáveis pelo encaminhamento, processamento e resolução de vários problemas do foro cívico, como iluminação pública, jardins, veículos abandonados, recolha de eletrodomésticos danificados, entre outros [Agência para a Modernização Administrativa, 2009a].

Lançado em 29 de Setembro de 2009 sobre a alçada do projeto Simplex 09, arrancou com as autarquias piloto de Arganil, Borba, Évora, Murça, Ovar, Pombal e Portalegre, às quais se juntaram grande parte dos restantes municípios de todos os distritos do país (em Fevereiro de 2012 segundo a Rede Comum de Conhecimento aderiram um total de 84 Municípios e 32 Juntas de Freguesias) [Agência para a Modernização Administrativa, 2008]. A ferramenta *Web* para notificação de problemas urbanos está acessível através do Portal do Cidadão³, constituindo um dos muitos serviços que os utentes têm ao seu dispor nesta plataforma [Agência para a Modernização Administrativa, 2008].

Apesar de minimalista, no sistema apresentado destacam-se:

- A integração com outros serviços cívicos do Portal do Cidadão (visibilidade e facilidade de acesso);
- Adaptação à realidade Portuguesa (representação fiel dos distritos, concelhos e freguesias que aderiram ao projeto, categorização de ocorrências baseadas em necessidades reais);
- Preenchimento minimalista de nova ocorrência (apenas texto, ou enriquecido com uma fotografia);
- Possibilidade do acompanhamento dos casos pelos cidadãos e também pelas autarquias.

Atingindo a marca das 9500 notificações enviadas em Fevereiro de 2012 [Agência para a Modernização Administrativa, 2008], a plataforma tem sido alvo de remodelações e melhorias, como é o caso da inclusão da tecnologia Google Maps para visualização num mapa do local da ocorrência.

³ Portal do Cidadão – Forma preferencial de acesso aos serviços disponibilizados pela Administração Pública de Portugal [Agência para a Modernização Administrativa, 2013]

2.2.6 CivicTicket

O CivicTicket apresenta-se como uma aplicação móvel de *crowd-sourcing* para reportar problemas cívicos que perturbem o dia-a-dia dos cidadãos [CivicTicket, 2012]. Com a missão de potenciar os seus utilizadores a reportarem as mais variadas situações (relacionadas com problemas da sua área de residência), este projeto oriundo do Sudeste Asiático conta já com a adesão de cidades como Kuala Lumpur, Petaling Jaya, Subang Jaya e Sepang. Atualmente, encontra-se numa fase de piloto de testes nas cidades de Damansara Kim e Kampung Damansara, também situadas na Malásia [CivicTicket, 2012].

Como um verdadeiro sistema de *crowd-sourcing*, este projeto centra o seu valor na comunidade e na participação ativa dos seus utilizadores, dependendo destes para alimentar a plataforma. Não é publicitado nenhum sistema para gestão das ocorrências por parte das entidades locais responsáveis pelo seu tratamento, delegando sobre a própria aplicação móvel o papel de ponte entre a comunidade e as instituições públicas.

Com o CivicTicket, os cidadãos reportam problemas urbanos através da aplicação móvel, que os torna públicos para outros utilizadores ou entidades de gestão local, atuando como um difusor de informação georreferenciada. Do ponto de vista do seu funcionamento, os cidadãos podem reportar um determinado problema de uma forma simplista, bastando tirar uma fotografia, descrever a ocorrência, marcar a sua localização e opcionalmente indicar a categoria em que esta se enquadra [CivicTicket, 2012]. Segundo [RGK Center, 2012], os relatórios enviados sobre a mesma ocorrência por utilizadores distintos são detetados e calculada a sua prioridade em tempo real.

Embora este projeto se encontre ainda no processo de desenvolvimento e teste contínuo, é possível retirar as seguintes ideias-chave [Jhet, 2012]:

- Forte aposta no conceito de *crowd-sourcing* – total dependência da comunidade para alimentar a plataforma;
- Sistema de conceção simples (assente numa arquitetura cliente-servidor, em que os clientes são apenas dispositivos móveis);
- Formulário de preenchimento de uma ocorrência minimalista;
- Apuramento de relatórios duplicados de ocorrências;
- Priorização de ocorrências seguindo vários parâmetros (votos, tempo até à resolução, etc.).

Atualmente, apenas está disponível a versão Android deste projeto, mas a equipa do CivicTicket anunciou já esforços para o desenvolvimento da versão iOS da aplicação.

2.2.7 Snap Send Solve

Snap Send Solve é o nome de um projeto Australiano de notificação de problemas urbanos desenvolvido pela Outware Mobile, empresa dedicada ao desenvolvimento de aplicações para os principais sistemas operativos móveis: iOS, Android e Windows Phone [Outware Mobile, 2013b]. Com distinções já atribuídas em vários concursos de aplicações móveis, a empresa aposta fortemente no desenvolvimento rápido de *software* que traga valor para as entidades governamentais, empresas, negócios locais e até organizações sem fins lucrativos, estando presente em vários contextos e segmentos do mercado, como aplicações cívicas, sociais, médicas e recreativas [Outware Mobile, 2013c].

O Snap Send Solve é anunciado pela empresa como uma das aplicações de referência na área de notificação de problemas urbanos, e o principal sistema no seu país de origem (Austrália) [Outware Mobile, 2013a]. Adaptado à realidade dos cidadãos australianos, esta aplicação está apenas disponível para os utilizadores daquele país, proporcionando-lhes uma experiência visual rica no envio de relatórios de ocorrências [Melbourne Design Awards, 2013]. Disponível nas plataformas iOS e Android, este projeto possibilita aos seus utilizadores o envio, visualização e partilha de relatórios georreferenciados, permitindo à semelhança de outros sistemas categorizar, acrescentar notas e fotografias para enriquecer o relatório. O Snap Send Solve foi recentemente alvo de reformulações profundas, tendo a equipa de desenvolvimento redesenhado e reconstruído de raiz a aplicação para dar resposta às críticas e sugestões dos seus utilizadores.

Agora na versão 2, o Snap Send Solve regista um largo número de melhorias em relação à versão anterior, das quais se realçam:

- A interface com o utilizador foi reconstruída para seguir um modelo passo-a-passo;
- Adicionado o suporte ao envio de múltiplas fotografias por relatório;
- Utilizar a localização atual do utilizador ou as coordenadas da fotografia;
- Enviar o relatório para várias entidades responsáveis em simultâneo;
- Visualizar entidades responsáveis de uma localidade, assim como os relatórios submetidos ou guardados em rascunho pelo utilizador, etc. [Outware Mobile, 2013d].

Embora se denotem estas melhorias na versão iOS da aplicação, não foi possível comprovar estas alterações na versão Android da aplicação. Porém, apesar de se mostrar um projeto de sucesso no seu país de origem, esta aplicação apresenta algumas falhas que outros sistemas já retificaram, como por exemplo: a notificação junto das entidades responsáveis é feita através de um e-mail (a aplicação automatiza a criação de um relatório na forma de e-mail para o utilizador), o que dificulta o processamento e triagem por parte da entidade [Foth et al., 2011].

2.2.8 PDX Reporter

PDX Reporter é o resultado de um esforço efetuado pelos órgãos de gestão governamental da cidade de Portland, no estado de Oregon nos Estados Unidos da América, para promover a troca de informação sobre os problemas da região e revolucionar os moldes atuais de interação entre as entidades de gestão e os cidadãos [Civic Apps, 2010]. Considerada uma das cidades mais preocupadas com o meio ambiente e com a sustentabilidade ecológica do Mundo, a cidade de Portland acrescenta o projeto PDX Reporter à sua panóplia de soluções para enriquecer a qualidade de vida dos seus cidadãos, possibilitando-lhes agora a notificação de problemas urbanos relacionados com o que julguem carecer de atenção por parte das entidades responsáveis pela sua resolução.

Através de um dispositivo iOS ou Android, os utilizadores do PDX Reporter podem submeter relatórios de problemas que detetem na sua comunidade local, como buracos nas estradas, iluminação pública danificada, mobiliário urbano danificado, manutenção de parques e paredes com grafiti, documentando o relatório com uma fotografia, localização geográfica da ocorrência, e uma breve descrição do ocorrido. A submissão do relatório é feita diretamente para o departamento responsável pela sua resolução (o departamento é determinado através da localização e tipo de ocorrência), e todos os passos do processo de resolução do problema são registados na aplicação, permitindo ao cidadão conhecer a qualquer momento o estado de resolução do problema [The City of Portland, 2013].

Muito embora esta aplicação esteja talhada para a realidade dos cidadãos de Portland, é possível extrair os seguintes pontos fundamentais da solução:

- A interface gráfica simplista, iniciando-se imediatamente pelo ecrã de nova notificação;
- Registo de notificações enviadas pelo utilizador no dispositivo;
- Categorização fixa e rápida do relatório, através de seleção numa caixa de opções;
- Encaminhamento automático da ocorrência para o departamento de Portland responsável;
- Código fonte disponível sobre a licença GPL versão 2.

Numa autêntica iniciativa de resolução de problemas locais com origem na comunidade e para benefício da comunidade, todo o código fonte da aplicação nas duas plataformas existentes (iOS e Android) está disponível nos repositórios de código livre da Google sobre a licença GPL, facilitando o acesso e cooperação com a equipa de desenvolvimento para fazer crescer esta solução [The City of Portland, 2013].

2.3 Análise comparativa das soluções de notificação de problemas urbanos identificadas

Todos os projetos supramencionados foram alvo de uma comparação a fim de determinar não apenas os seus pontos comuns, mas também para elucidar as suas divergências, por forma a se poder estabelecer um conjunto de funcionalidades base, e corroborar a necessidade de algumas características inovadoras a suportar na solução proposta neste documento. A relevância desta comparação, ao invés de se focar estritamente na comparação parâmetro a parâmetro com vista à escolha da melhor solução, prende-se no apuramento dos requisitos dados como adquiridos pelos utilizadores de sistemas de notificação de problemas urbanos, e pela descoberta de novos que consigam alavancar a motivação dos cidadãos na utilização do sistema a desenvolver. A Tabela 1 resume a confrontação efetuada entre os sistemas, reunindo somente os parâmetros que apresentam maior disparidade entre as várias soluções:

Tabela 1 — Notificação de problemas urbanos: Comparação dos projetos analisados

	FixMyCity	FixMyStreet	CitySourced	SeeClickFix	A minha Rua	Civic Ticket	Snap Send Solve	PDX Reporter
Tem uma aplicação para os vários S.O Móveis?	Sim	Não oficialmente	Sim	Sim	Não	Sim	Sim	Sim
Requer autenticação para reportar ocorrências?	Sim	Sim, confirmação de <i>e-mail</i>	Sim	Sim	Sim	Sim	Sim, requer envio de e-mail	Não
Vários tipos de anexos multimédia?	Não, apenas foto	Não, apenas foto	Sim, multimédia	Sim, multimédia	Não, apenas foto	Não, apenas foto	Não, apenas foto	Não, apenas foto
Auxilia na categorização das situações reportadas?	Sim, por caixa de seleção	Sim, por caixa de seleção	Sim, por caixa de seleção	Sim, por caixa de seleção e módulo Q&A	Sim, por caixa de seleção	Sim, por caixa de seleção	Sim, por caixa de seleção	Sim, por caixa de seleção
Possui aplicação Web?	Sim	Sim	Sim	Sim	Sim	Não	Sim, para as entidades municipais	Não
Acessível para todo o mundo?	Ainda não divulgado	Adaptável	Sim	Sim	Não, apenas Portugal	Não, apenas Malásia	Não, apenas Austrália	Não, apenas Portland, EUA
Livre de utilização pelos cidadãos?	Ainda não divulgado	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Livre de utilização pelos municípios?	Não	Sim	Não	Não	Sim	Sim	Sim	Não, apenas Portland

Implementa o standard Open311?	Não	Sim	Sim	Sim	Não	Não	Não	Não
Qual o tipo de licenciamento de software?	Licença proprietária	AGPL v3	Licença proprietária	Licença proprietária	Licença proprietária	Licença proprietária	Licença proprietária	GPL v2
Efetua o encaminhamento para as entidades responsáveis?	Sim, via <i>backoffice</i>	Sim, via <i>backoffice</i>	Sim, via <i>backoffice</i>	Sim, via <i>backoffice</i>	Sim, via <i>backoffice</i>	Sim	Sim, via <i>e-mail</i>	Sim, via <i>backoffice</i>
Acompanhamento do estado de resolução da ocorrência?	Sim, via aplicações móvel e web	Sim, via aplicações móvel e web	Sim, via aplicações móvel e web	Sim, via aplicações móvel e web	Sim, via aplicações móvel e web	Sim, via aplicação móvel	Sim, via aplicação móvel ou e-mail	Sim, via aplicação móvel
Receção de notificações de outros utilizadores em contexto georreferenciado?	Sim, via aplicação móvel num raio de pesquisa	Sim, via aplicação <i>web</i> por RSS ou <i>e-mail</i>	Sim, por uma área selecionada pelo utilizador	Sim, por uma área selecionada pelo utilizador	Não	Sim, via aplicação móvel num raio de pesquisa	Não	Sim, via aplicação móvel num raio de pesquisa
Informação da localização nos relatórios?	Sim	Sim	Sim, com indicação da orientação do dispositivo	Sim	Sim	Sim	Sim	Sim
Suporta discussão ou comentários nos relatórios?	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Efetua priorização das ocorrências?	Sim, baseado nas redes sociais	Não	Sim, através de votos	Sim, através de votos	Não	Sim, através de votos	Não	Não
Difusão das ocorrências nas redes sociais?	Sim	Não	Sim	Sim	Não	Não	Não	Não

Os parâmetros utilizados nesta comparação foram determinados através de uma análise transversal a todos estes sistemas, permitindo extrair apenas as características que definem cada uma das soluções, e descartar as que se repetem. Estas últimas serão reaproveitadas como funcionalidades base do sistema a desenvolver, uma vez que, na perspetiva dos utilizadores, são encaradas como os requisitos mínimos neste tipo de sistemas.

Embora a designação dos critérios seja algo autoexplicativa, a linguagem utilizada pode estar fora do âmbito do leitor, pelo que se torna necessário elucidar cada um dos parâmetros:

- “Tem uma aplicação para os vários S.O. Móveis?”: indica se o projeto tem ou não uma componente móvel;
- “Requer autenticação para reportar ocorrências?”: refere se o sistema requer algum tipo de autenticação para reportar ocorrências;
- “Vários tipos de anexos multimédia?”: permite saber se as aplicações suportam vários elementos multimédia para enriquecer o relatório (áudio, vídeo e imagem) além da típica fotografia;
- “Auxilia na categorização das situações reportadas?”: informa se os sistemas auxiliam o utilizador no momento de categorização da ocorrência;
- “Possui aplicação *Web*?”: indica se o projeto inclui uma componente *Web*;
- “Acessível para todo o mundo?”: diz se o projeto está difundido internacionalmente e é passível de ser utilizado;
- “Livre utilização pelos cidadãos?”: refere se o sistema é de livre utilização (sem custos associados) pelos cidadãos;
- “Livre utilização pelos municípios?”: informa se o sistema está isento de pagamento por parte dos municípios que o adotem;
- “Implementa o *standard* Open311?”: averigua se o projeto cumpre as diretivas estabelecidas pela norma Open311;
- “Qual o tipo de licenciamento do software?”: indica o tipo de licenciamento a que o projeto está sujeito (código aberto/fechado);
- “Efetua o encaminhamento para as entidades responsáveis?”: refere se o sistema é capaz de encaminhar o relatório da ocorrência diretamente para a entidade responsável;

- “Acompanhamento do estado de resolução da ocorrência?”: informa se a evolução do estado de resolução de uma determinada ocorrência é registrado e disponibilizado ao utilizador que reportou a situação;
- “Receção de notificações de outros utilizadores em contexto georreferenciado?”: refere se os utilizadores recebem as mensagens considerando a sua localização num determinado instante;
- “Informação da localização nos relatórios?”: indica se os relatórios de ocorrências têm associadas as coordenadas do local a que se referem;
- “Suporta discussão ou comentários nos relatórios?”: diz se o projeto comporta um módulo de discussão ou adição de comentários a relatórios de ocorrências;
- “Efetua priorização das ocorrências?”: informa se o sistema disponibiliza um mecanismo para priorizar a resolução de ocorrências;
- “Difusão das ocorrências nas redes sociais?”: indica se os relatórios de ocorrências podem ser partilhados nas redes sociais;

2.4 Conclusões

Da realização deste estudo podem-se retirar várias conclusões proeminentes relacionadas com sistemas de notificação de problemas urbanos. Por um lado, realçou-se a crescente importância deste tipo de sistemas no seio das comunidades locais, enfatizando a sua contribuição para melhorar substancialmente a qualidade de vida de zonas urbanas, auxiliando tanto os cidadãos como as autarquias na resolução de problemas cívicos não urgentes. A redução de custos de operação e otimização dos processos de deteção e resolução também foi alvo de atenção neste capítulo, demonstrando o impacto que estes sistemas de apoio provocam no balanço económico dos municípios, considerando a instabilidade económico-financeira que se vive na sociedade contemporânea. Por outro lado, foi efetuado um levantamento do estado da arte deste ramo de aplicações, incluindo os sistemas do mercado que têm mais valor e notoriedade e que representam a maior parte dos cenários possíveis de implementação e utilização. Ainda neste capítulo, as soluções apresentadas foram alvo de uma análise pormenorizada das suas capacidades, e obtidos os pontos em que coincidem e se distinguem, a fim de se estabelecer uma base comum para este tipo de sistemas, e determinar as características que definem cada um dos sistemas e os tornam atrativos para os utilizadores. Findo este levantamento, as ferramentas foram confrontadas com um conjunto de critérios bem definidos para avaliação dos seus pontos divergentes, e delineação de funcionalidades inovadoras que resultem em novas oportunidades de negócio nesta área, e que se traduzam em valor acrescentado para os cidadãos.

Deste trabalho de pesquisa e avaliação intensivo, foi possível consensualizar que as características mais comuns nestes sistemas e que correspondem ao mínimo indispensável para os utilizadores passam pela disponibilização de uma aplicação móvel e *Web* para criação de relatórios de ocorrências, com suporte ao envio de uma descrição da ocorrência, uma fotografia para ilustrar o acontecido, a localização geográfica da situação e uma forma de categorizar a ocorrência, para ser posteriormente encaminhada para o departamento responsável. A visualização de ocorrências num determinado raio ou área também se mostrou habitual, assim como o acesso pelos municípios a um *backoffice* para gestão das ocorrências reportadas.

A análise efetuada permitiu ainda detetar algumas fragilidades presentes em todos os sistemas, e assim esboçar algumas funcionalidades que poderão motivar os cidadãos na utilização da ferramenta a desenvolver. A fraqueza primordial destes sistemas, e que origina toda a discussão levantada por esta tese, é a categorização simplista das ocorrências. Este tipo de categorização serve-se tipicamente de caixas de seleção (como é o caso da maioria dos sistemas) ou são também complementadas com um sistema de perguntas e respostas (como é o caso específico do SeeClickFix). Apesar de eficaz, este tipo de categorização não se revela eficiente, pelo que leva o utilizador a perder tempo na seleção de uma categoria (em condições normais existem pelo menos 5 categorias). A equipa de desenvolvimento encarou esta debilidade como uma oportunidade, e decidiu avançar para o desenvolvimento de um sistema de categorização automática das ocorrências mais comuns, servindo-se para isso do

reconhecimento de padrões em imagem (ver capítulo 4). Outra das fragilidades encontradas foi a falta de filtro na informação enviada às entidades responsáveis. Por um lado, estas aplicações de foro cívico eliminam a necessidade de intermediários (tipicamente uma linha telefónica dedicada, [Foth et al., 2011]), e a informação relevante é inserida imediatamente nos sistemas das entidades de suporte, mas com a ausência de intermediários para eliminarem informação desnecessária, essa tarefa recai sobre os funcionários dessas entidades. Um dos cenários vulgares é a notificação de uma ocorrência utilizando apenas o nome e uma imagem, sem clarificar na fotografia qual o problema identificado. Mais uma vez, esta característica foi encarada como uma possibilidade de melhoria pela equipa deste projeto, que resolveu avançar para o desenvolvimento de um módulo para demarcação manual de objetos em fotografias das ocorrências (ver capítulo 3). Desta forma, os utilizadores poderão demarcar o problema nas fotografias das ocorrências, eliminando erros de interpretação por parte das entidades municipais.

Foi ainda possível efetuar uma avaliação pragmática das ferramentas, evidenciando a supremacia das soluções SeeClickFix e CitySourced em relação às restantes, pelo seu grau de maturidade e pela adesão da comunidade, e pela sua forte contribuição para a definição dos *standards* nesta área [Open311, 2013b]. Embora todas tenham características que as demarquem e uma área de influência bem definida, que atrai um núcleo próprio de utilizadores, o projeto SeeClickFix apresenta-se como o mais completo e reconhecido na comunidade.

3 Anotações em imagem

3.1 Introdução

Uma anotação refere-se tipicamente ao processo de associação de metadados⁴ [Shih et al., 2012] que classifiquem, categorizem ou identifiquem um determinado objeto. No caso específico das imagens, a informação está representada sobre a forma de palavras-chave ou legendas na maioria dos casos [Shih et al., 2012], mas através da utilização de métodos auxiliares é possível embeber uma vasta panóplia de objetos, tais como pontos, linhas, polígonos, entre outros [LEAD, 2013c].

A investigação no ramo das anotações em imagem é extensa, principalmente quando se trata da automatização deste processo, em que se propõem diversas técnicas e algoritmos para atingir o melhor compromisso entre o desempenho e a fiabilidade desejados [Hu & Lam, 2013]. Segundo Chen [Chen et al., 2012], a anotação de imagens sem intervenção de um ator humano aplica-se nos casos em que existe um elevado conjunto de dados a catalogar, que a ser processado manualmente obrigaria a um esforço laboral intensivo e demorado, e estaria naturalmente sujeito à ocorrência de gafes ou erros de interpretação.

No âmbito deste projeto, o volume de informação a processar será repartido pelos cidadãos ativos da comunidade, que ao reportarem ocorrências e demarcarem as zonas problemáticas estarão a contribuir para a correta anotação dessas imagens, e simultaneamente estarão a facilitar a função de triagem das organizações responsáveis pela sua resolução. Assim sendo, a inclusão de um módulo para a demarcação manual de problemas nas imagens no sistema a desenvolver servirá o propósito de anotação das fotografias reportadas, aproveitando as potencialidades do *crowd-sourcing* para delegar o trabalho normalmente executado pelos funcionários das entidades municipais nos cidadãos das comunidades locais. Para aliciar os cidadãos na utilização desta funcionalidade, o módulo a desenvolver terá de adotar um modo

⁴ Metadados – De uma forma simplista, metadados representam informação acerca dos dados: permitem descrever e caracterizar conteúdo [Chuttur, 2011].

de operação que se baseie numa interface natural para os utilizadores de dispositivos móveis, e ser rápido e eficiente para agilizar ao máximo a tarefa da demarcação nas imagens.

No capítulo anterior, foram delineadas algumas funcionalidades inovadoras a desenvolver na solução de notificação de problemas urbanos proposta nesta tese. Este capítulo procura descrever uma dessas inovações: a demarcação manual de uma área em imagem, um subtipo de anotações possíveis de serem efetuadas em imagem. Feita uma breve contextualização da ideia a desenvolver, irão ser apresentadas algumas aplicações que têm sucesso nesta matéria, e posteriormente identificadas as principais soluções encontradas para a implementação ou integração de uma técnica de demarcação manual em imagens.

3.2 Aplicações existentes

A implementação de uma técnica para demarcação manual de áreas em imagem pode ser atingida seguindo abordagens distintas: uma das possíveis é a adoção de anotações em imagem, em que tipicamente existe informação adicional (metadados) associada a uma fotografia, que é interpretada e processada aquando da apresentação da imagem ao utilizador (através da adição de novas camadas sobre a imagem); também pode ser concretizada recorrendo à simples manipulação das imagens originais, que resultam em novas imagens que refletem as alterações efetuadas.

A fim de elucidar as diferenças entre estas duas abordagens, nesta secção irão ser apresentados alguns exemplos de aplicações que implementam as duas vias de demarcação de áreas em imagem supramencionadas em dispositivos com sistema operativo *Android*.

A primeira solução a ser apresentada tem o nome de **Skitch** (ver Figura 1), disponível para dispositivos móveis Android e iOS e as plataformas *desktop* Windows e Mac OS X. Desenvolvido pela empresa Evernote, criadora de produtos de sucesso como Evernote, Penultimate e Web Clipper, o Skitch é a aposta da empresa na área de criação de anotações, formas e esboços em documentos e imagens, constituindo uma das aplicações de referência em plataformas móveis [Evernote Corporation, 2013].



Figura 1 – Skitch - Adição de uma anotação numa imagem

Com atualizações frequentes publicadas no Google Play, esta solução de anotações com instalação gratuita é das mais populares, contando com um número de instalações superior a 5 milhões [Google, 2013c].

Editor de Imagem (ver Figura 2) é o nome de um projeto que se dedica à manipulação de imagens, apresentando-se como a alternativa móvel às versões *desktop* dos pacotes de aplicações GIMP e Photoshop. Numa abordagem distinta da selecionada pela aplicação Skitch, esta solução foca-se no suporte a múltiplas camadas, e operações simples sobre imagens como redimensionar, cortar, apagar, e desenhar formas, relevando-se uma aplicação completa para os utilizadores que desejam efetuar alterações a fotografias imediatamente após a sua captura [Byte Experts, 2013].

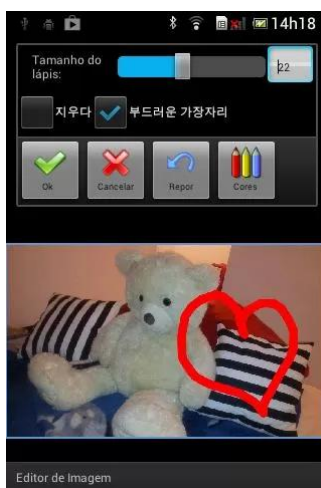


Figura 2 – Editor de imagem – Edição de uma fotografia [Google, 2013b]

O **Photoshop Touch** (ver Figura 3), a versão móvel do aclamado *software* de edição de imagem Photoshop® da gigante tecnológica Adobe®, apresenta-se como um concorrente de peso na categoria de manipulação de imagem, importando grande parte dos conceitos e funcionalidades da versão original para *desktops*.



Figura 3 – Adobe Photoshop Touch – Manipulação de uma fotografia

3.3 Tecnologias de anotação em imagem

3.3.1 LEADTOOLS - Document Image Annotations

LEADTOOLS é o nome de um pacote de ferramentas de desenvolvimento publicado pela Lead Technologies, empresa dedicada à distribuição de soluções para tratamento de imagem [LEAD, 2013a]. Com vários *SDKs* (*kits* de desenvolvimento de software) destinados a ramos específicos da indústria, como multimédia, medicina e fotografia, o LEADTOOLS disponibiliza um pacote especializado para anotações em documentos e imagens: o LEADTOOLS Image Annotation SDK (ver Figura 4) [LEAD, 2013b].



Figura 4 – LEADTOOLS - Anotações em documentos

Este pacote oferece funções específicas para adição de anotações em imagens, documentos e formatos de imagem médicos (por exemplo, DICOM), permitindo embeber diretamente as anotações nas imagens originais e consequentemente gerar novas imagens, ou guardadas numa camada de apresentação [LEAD, 2013b]. Suporta uma vasta coleção base de objetos passíveis de serem inseridos nas imagens, como cliques de áudio, botões, pontos, linhas e polígonos, elipses, setas, etc. [LEAD, 2013c], permitindo ainda ao programador estender esta coleção com objetos feitos à medida que deseje representar. O armazenamento das anotações é flexível e pode ser feito recorrendo a ficheiros separados das imagens originais, bases de dados, memória ou embebidos diretamente nos ficheiros originais. A facilidade de uso e controlo está presente tanto na perspetiva do programador como na perspetiva do utilizador final, sendo extensível a vários níveis para tornar a experiência dos utilizadores o mais rica possível [LEAD, 2013b]. Muito devido à aplicabilidade desta tecnologia no ramo da medicina, a confidencialidade e a segurança não foi esquecida pela equipa responsável pelo LEADTOOLS. [LEAD, 2013b]. Com o uso de objetos para encriptar partes de documentos ou imagens, é possível definir áreas de acesso num documento ou imagem e atribuir permissões de visualização diferentes aos utilizadores.

O pacote necessário para o desenvolvimento de uma solução com potencialidades de anotação em imagem tem um custo demasiado elevado para o orçamento do projeto, invalidando a sua utilização.

3.3.2 Android Graphics 2D API

Outra alternativa para a implementação de um módulo de demarcação de áreas em imagem passa pela utilização das APIs de manipulação de gráficos 2D disponibilizadas por bibliotecas nativas do sistema operativo *Android*.

Para além das bibliotecas de alto nível para inserção e representação de controlos de interface com o utilizador mais comuns como entradas de texto, imagens, botões, entre outros, o sistema operativo *Android* exporta uma framework para programação em que é possível estender a plataforma segundo as necessidades de utilizadores e programadores [Samsung, 2012]. Uma das bibliotecas constituintes da framework de base do *Android* é a 2D Drawing API, que permite aos programadores adicionar novos elementos à interface gráfica e tomar controlo das fases de renderização e interação desses mesmos componentes. Este conjunto de APIs pode ser utilizado para construir um novo componente de interface que reaja à delineação que o utilizador irá realizar no objeto a realçar, e permita gerar uma nova imagem que contenha as marcas efetuadas [Google, 2013a].

De acordo com a documentação oficial da API de gráficos do *Android*, existem duas possibilidades para o desenvolvimento de controlos que utilizem potencialidades 2D: utilizar o objeto padrão “View” para reaproveitar ao máximo os eventos da interface gráfica por omissão do sistema operativo, ou utilizar um objecto “Canvas” que requer que alguma lógica de apresentação e criação de elementos 2D esteja a cabo do programador.

A Figura 5 ilustra as potencialidades desta API, desenhando várias linhas num objeto do tipo “Canvas” que é atualizado periodicamente [Google, 2013a]:

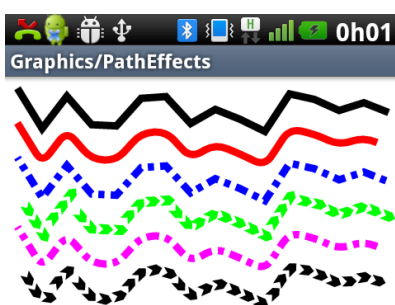


Figura 5 – Drawing API – Demonstração de potencialidades

Esta solução não carece de qualquer pagamento ou restrição para a sua utilização, e é distribuída como parte integrante da plataforma Android sobre a licença de código livre Apache Software License versão 2.0.

3.4 Análise comparativa das tecnologias de anotação estudadas

As alternativas propostas neste capítulo foram minuciosamente analisadas e colocadas lado a lado para comparação, a fim de se poder determinar qual a mais viável para o projeto a desenvolver. Na realização deste estudo, foram também identificadas várias bibliotecas de tratamento de imagem para os sistemas *Android* (*droidreader*, *jpgdfnotes*, *jjil*, entre outras), que se limitam a incluir novas camadas sobre a API de desenho deste sistema operativo, apresentando otimizações para áreas de negócio bem definidas. No âmbito deste projeto, não foi encontrada nenhuma mais-valia na utilização dessas bibliotecas, uma vez que iriam gerar complexidade e necessidade de manutenção das mesmas durante o ciclo de vida da aplicação. A Tabela 2 reúne o resultado da comparação efetuada entre as duas principais soluções identificadas:

Tabela 2 — Tecnologias de anotação - Comparação das soluções encontradas

	LEADTOOLS Document/Image Annotations	Android Drawing API
Suporte nativo a anotações?	Sim	Não
Possibilidade de incluir as anotações no ficheiro de imagem original?	Sim	Não
Geração de ficheiros de descrição de anotações?	Sim	Não
Suporte ao desenho livre de formas sobre imagens?	Sim	Sim
Tipo de licença?	Paga	Livre (Apache License 2.0)

Os parâmetros utilizados para realizar esta comparação focaram-se no domínio das anotações em imagem, descurando a potencialidade destes sistemas para realizar outro tipo de operações. Devido às restrições orçamentais do projeto, foi também considerado o tipo de licenciamento praticado em cada uma das soluções.

Para uma melhor compreensão da análise efetuada, de seguida são descritos todos os parâmetros que compõem a grelha de comparação:

- “Suporte nativo a anotações?”: indica se a solução suporta de base a inclusão de anotações em imagens;
- “Possibilidade de incluir as anotações no ficheiro de imagem original?”: refere se a alternativa permite a inclusão das anotações diretamente no ficheiro original, sem recurso a ficheiros adicionais (tipicamente através da geração de uma nova imagem);

- “Geração de ficheiros de descrição de anotações?”: indica se a API é capaz de gerar ficheiros que descrevam as anotações criadas nas imagens, mantendo a imagem original inalterada;
- “Suporte ao desenho livre de formas sobre imagens?”: refere se a plataforma possibilita o desenho livre de formas pelo utilizador, sobre as imagens que se pretendem;
- “Tipo de licença?”: descreve o tipo de licenciamento a que cada uma das soluções está sujeita.

3.5 Conclusões

O estudo realizado sobre anotações em imagem, sumarizado neste capítulo, permitiu identificar, avaliar e optar por uma das soluções para a implementação de uma técnica de demarcação manual de imagens de forma sustentada, justificando a escolha com base na análise pormenorizada das duas principais alternativas encontradas.

Foi também alvo deste estudo verificar, numa primeira fase, que aplicações do mercado implementam com sucesso este tipo de técnicas de demarcação em imagens. Feito o levantamento das aplicações proeminentes nesta área, e dos métodos que utilizam para possibilitar a demarcação manual em imagens, foram discutidas duas abordagens possíveis de adotar neste projeto: a utilização de um SDK da LEADTOOLS, otimizado para realizar operações sobre imagens, e a API para gráficos 2D da plataforma *Android*. Analisadas as vantagens e desvantagens de cada uma das alternativas, e considerando todas as condicionantes do projeto, é possível concluir que a opção de implementar a técnica de demarcação manual em imagens recorrendo à API de gráficos 2D da plataforma *Android* é a melhor solução, uma vez que não carece de qualquer pagamento para a sua utilização e, apesar de requerer mais esforço por parte da equipa de desenvolvimento para implementar o mecanismo de anotações, é a opção que apresenta a melhor relação custo/benefício para o projeto. Embora o quadro-resumo da comparação efetuada entre as alternativas mostre uma clara vantagem para a solução oferecida pela LEADTOOLS, nesta tomada de decisão pesou sobretudo a disponibilidade orçamental limitada do projeto, que invalida por completo o SDK da LEADTOOLS.

A equipa do projeto irá então utilizar as bibliotecas de desenho 2D disponibilizadas pelo *Android*, para desenvolver um componente que permita a demarcação manual pelo utilizador de imagens em que seja necessário realçar o problema a reportar.

4 Reconhecimento de padrões em imagem

4.1 Introdução

Um padrão, segundo Watanabe [Watanabe, 1985], caracteriza-se como “uma entidade, vagamente definida, à qual pode ser atribuído um nome”, como por exemplo uma imagem de uma impressão digital, uma palavra escrita com caligrafia humana, um rosto ou um sinal de voz [Jain et al., 2000]. O reconhecimento de padrões apresenta-se como a área científica responsável pelo estudo de como sistemas computadorizados conseguem observar o ambiente, distinguir padrões de interesse e tomar decisões sobre a informação constante nos objetos identificados [Jain et al., 2000]. Esta capacidade de reconhecer modelos e padrões, constitui uma das tarefas do cotidiano dos seres humanos, grande parte das vezes levada a cabo instintivamente. No entanto, transferir este poder de análise e depreensão para máquinas e sistemas informáticos representa um grande desafio para a comunidade científica, que continua sem uma resposta concreta para as necessidades dos vários ramos da ciência e da indústria, como a biologia, psicologia, medicina, marketing, computação visual, inteligência artificial, entre outros [Jain et al., 2000]. A automatização dos processos de reconhecimento requer que os sistemas sejam capazes de identificar, descrever, classificar e agrupar padrões, implementando um de dois métodos de reconhecimento e classificação: classificação supervisionada ou classificação não supervisionada [Jain et al., 2000]. A classificação supervisionada, técnica muito utilizada no campo da aprendizagem de sistemas (um ramo da inteligência artificial), utiliza um conjunto de dados representativo do padrão a identificar, conhecido como conjunto de dados de treino. Devido à utilização de informação fiável sobre os objetos do domínio a capturar, esta técnica tende a devolver resultados mais precisos do que a classificação não supervisionada [Chaovalit & Zhou, 2005]. Embora retorne numa primeira instância melhores resultados do que a classificação não supervisionada, a utilização de um conjunto de dados de treino foca o reconhecimento num contexto específico, conjunto esse que necessita de ser novamente gerado quando aplicado num contexto distinto [Chaovalit & Zhou, 2005]. A classificação não supervisionada (também denominada *data*

clustering) identifica objetos com base na similaridade de padrões encontrados e atribui-lhes uma classe até então desconhecida [Ari & Aksoy, 2010]. Como o reconhecimento dos padrões desta técnica não recorre a modelos pré-determinados nem conjuntos de treino, toda a informação disponível está por classificar [Jain et al., 2000]. Esta técnica é comumente utilizada em situações onde é necessário encontrar agrupamentos (ou *clusters*) em dados multidimensionais, apenas com base na ponderação ou percepção do grau de semelhança entre padrões [Jain et al., 2000]. Estudos recentes demonstram que a combinação destas duas técnicas de classificação resulta num melhor reaproveitamento dos pontos fortes de ambas, através da utilização de informação já categorizada (por exemplo, os dados de um conjunto de treino) e não categorizada. Difundida como classificação semi-supervisionada, esta nova técnica tornou-se recentemente um tópico de especial interesse junto da comunidade científica, da qual já derivaram várias propostas de abordagens utilizando este método (por exemplo, *Self-Training*, *Co-Training* ou *Generative Models*) [Gan et al., 2012].

A aplicabilidade da teoria relacionada com o reconhecimento de padrões é comprovada pela implementação de sistemas e soluções para vários setores da sociedade. Áreas do conhecimento como a inteligência artificial, informática, medicina, arqueologia ou armamento militar, possuem aplicações no campo da computação gráfica, diagnóstico auxiliado por computador, imagiologia médica, reconhecimento de caracteres de caligrafia humana, reconhecimento de faces e de voz, análises de ADN, entre outras [Liu et al., 2006].

A temática abordada nesta investigação cinge-se ao estudo de um subgrupo de aplicações da área da computação gráfica (ou *computer vision*): o reconhecimento de padrões em imagem, ou simplesmente reconhecimento de imagem. Como conceito derivado do reconhecimento de padrões, o reconhecimento de imagem herda grande parte das suas características e técnicas, como classificação supervisionada (recorrendo a um conjunto de imagens de treino), semi-supervisionada e não supervisionada.

Este capítulo pretende introduzir a problemática do reconhecimento de padrões em imagem e contextualizá-la no projeto, começando por apresentar algumas aplicações para dispositivos móveis que implementam técnicas de reconhecimento e classificação. Segue-se o levantamento das principais tecnologias móveis existentes para processamento de imagem em ambiente móvel, passando para a sua confrontação com base num conjunto de parâmetros pré-definido. No final são retiradas algumas conclusões sobre o trabalho efetuado, rematando com a ponderação final sobre a avaliação efetuada.

4.2 Aplicações existentes

O estudo de sistemas para o reconhecimento de imagem seguiu a metodologia levada a cabo nas restantes temáticas abordadas nesta investigação, começando por efetuar um levantamento das aplicações existentes no mercado que se demarcam das restantes pelo seu sucesso e grau de inovação numa área de negócio específica. O levantamento efetuado neste capítulo resume a oferta disponível no mercado relativamente a aplicações com capacidades de reconhecimento de padrões em imagem, descrevendo um conjunto de soluções que constituam uma amostra representativa das várias alternativas existentes. Neste segmento de atividade, é comum o aparecimento de aplicações que usufruam essencialmente das potencialidades dos sensores embutidos nos dispositivos móveis, como o GPS, câmaras fotográficas de alta resolução ou bússolas [Casanova et al., 2013]. A presença destes e outros sensores nos dispositivos torna-os alvos preferenciais para o desenho e conceção de aplicações direcionadas para a computação gráfica, como sistemas de reconhecimento facial, reconhecimento de códigos de barras para o mercado retalhista, compra de bilhetes de cinema através da leitura dos cartazes dos filmes, reconhecimento de pontos de interesse para turistas, entre outros [Casanova et al., 2013]. O crescente número de aplicações de processamento de imagem é também justificado pela difusão e popularidade de plataformas móveis como *smartphones* e *tablets*, assim como a tendência para estes equipamentos melhorarem exponencialmente o seu desempenho, devido em grande parte ao aumento de recursos de processamento disponíveis, e à otimização dos sistemas operativos destes aparelhos.

4.2.1 Google Goggles

Numa tentativa de complementar a pesquisa textual que já é característica do gigante dos motores de busca Google, surge o projeto Goggles - um sistema de pesquisa visual - que utiliza técnicas de computação gráfica para reconhecer objetos em imagens enviadas pelos utilizadores, retornando toda a informação disponível sobre os objetos identificados [Handa, 2012].

Capaz de reconhecer marcos históricos, capas de livros, logótipos de lojas, ler códigos de barras de produtos e apresentar informação detalhada do produto, reconhecer os caracteres de um texto (ver Figura 6) e efetuar a sua tradução automática ou extrair os detalhes de um contacto num cartão profissional através de imagens, representa uma das aplicações para sistemas operativos *Android* e *iOS* que implementa grande parte das potencialidades de *computer vision* para dispositivos móveis. Com uma interface simples que convida à sua utilização, o Google Goggles não requer qualquer tipo de registo prévio à sua utilização, exigindo apenas que o utilizador tire uma fotografia ou selecione uma imagem da galeria para análise pelo sistema.

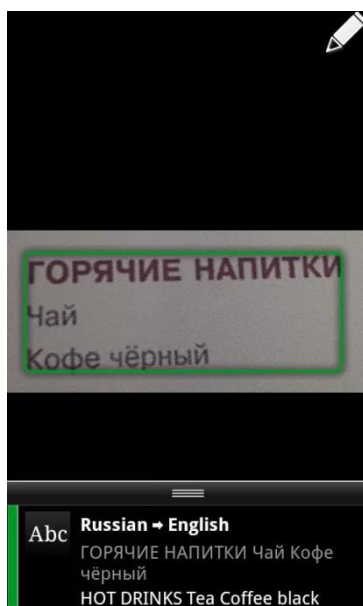


Figura 6 – Google Goggles – Reconhecimento de caracteres

Apesar de desenvolvido numa fase inicial do projeto, o reconhecimento de faces foi colocado de parte do Google Goggles por questões relacionadas com a privacidade dos utilizadores. Embora seja uma funcionalidade desejada pelos utilizadores (de acordo com um levantamento efetuado em 2010, cerca de 25% de todas as imagens submetidas para a plataforma contêm faces [Takahashi, 2010]), muito dos resultados encontrados involuntariamente podem ser utilizados para fins desconhecidos (uma fotografia de um indivíduo pode sugerir o seu blogue pessoal ou redes sociais). De acordo com o mesmo estudo, a versão ainda experimental do Goggles tem já uma eficácia de trinta e três por cento a

retornar apenas um resultado, que segundo o sistema corresponde exatamente à pesquisa efetuada [Takahashi, 2010].

Com o processamento mais complexo a cargo dos vários servidores e motores de pesquisa, a aplicação móvel limita-se a submeter as fotografias para a rede de sistemas da Google, retornando, em média, três resultados a cada 6,5 segundos [Takahashi, 2010]. Devido a estar assente numa arquitetura distribuída, a aplicação móvel requer uma ligação ativa à internet para efetuar as pesquisas [Chandler, 2012].

Para efetuar o reconhecimento de uma imagem, o sistema utiliza múltiplas técnicas: numa primeira instância, tenta identificar objetos na fotografia através de algoritmos de reconhecimento de padrões (como deteção dos cantos de uma zona de interesse, ou pontos chave extraídos de um conjunto de referência) [Handa, 2012] (ver Figura 7), recorrendo à base de dados de imagens interna do Google; se forem detetados caracteres na imagem (através de técnicas de OCR – Optical Character Recognition), estes são lidos e inseridos na parametrização da pesquisa, restringindo o leque de pesquisa; a posição geográfica da fotografia também é levada em conta, uma vez que se pode tratar de uma fotografia de um marco histórico nas imediações.

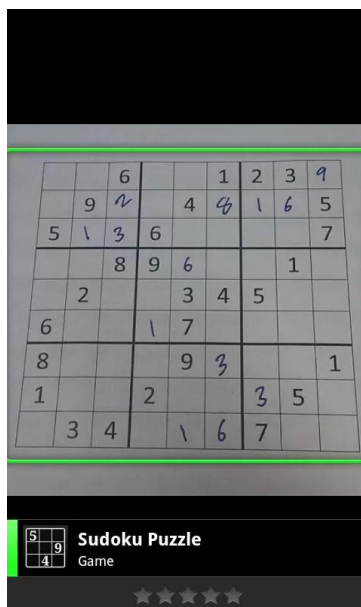


Figura 7 – Google Goggles – Reconhecimento de padrões e classificação

Embora ainda se encontre numa fase experimental, e com desenvolvimento contínuo por parte da equipa do Google Labs, este projeto conta já com mais de dez milhões de instalações no mercado de aplicações Google Play. A longo prazo, e acompanhando tanto a evolução da aplicação como das capacidades de processamento dos seus servidores, a empresa espera no futuro aplicar este conceito a várias áreas de negócio e contextos distintos, como transações financeiras e processamento de cartões de crédito, ou simplesmente detetar o estado de espírito do utilizador [Takahashi, 2010].

4.2.2 Visidon AppLock

AppLock é o nome de uma solução desenvolvida pela empresa Visidon para assegurar a privacidade dos utilizadores de dispositivos móveis, através do bloqueio do acesso às aplicações por reconhecimento facial do proprietário do aparelho [Visidon, 2013a]. O controlo total do bloqueio do dispositivo está disponível numa versão paga da aplicação no Google Play, sobre a forma de uma funcionalidade ainda em desenvolvimento. Também o desenho de um padrão ou o fornecimento de um PIN são formas alternativas de desbloqueio do ecrã presentes na versão paga, assim como a disponibilização de um controlo externo para aplicações de automação (como Tasker ou Locale) [Visidon, 2013a].

Para o utilizador usufruir da melhor experiência possível com a aplicação, é recomendada a utilização da versão *Gingerbread* ou superior do *Android*. Para o reconhecimento facial, é também necessário que o dispositivo tenha uma câmara fotográfica frontal. Devido à fragilidade inerente ao reconhecimento facial, e à existência de fatores que afetam gravemente a qualidade das fotografias, como pouca luminosidade, preservação das cores ou ruído, a configuração de um método alternativo de desbloqueio é aconselhada pela equipa da Visidon [Visidon, 2013a].

Assegurado o método alternativo de desbloqueio do sistema, o utilizador é convidado a treinar a aplicação, fornecendo para o efeito imagens com diferentes condições de luminosidade, contraste, e ruído (ver Figura 8). Após o treino do algoritmo de reconhecimento, é possível definir as aplicações que se desejam bloquear recorrendo ao reconhecimento facial (por exemplo o navegador de internet, aplicação de mensagens, galeria de imagens, etc.) [Visidon, 2013a].

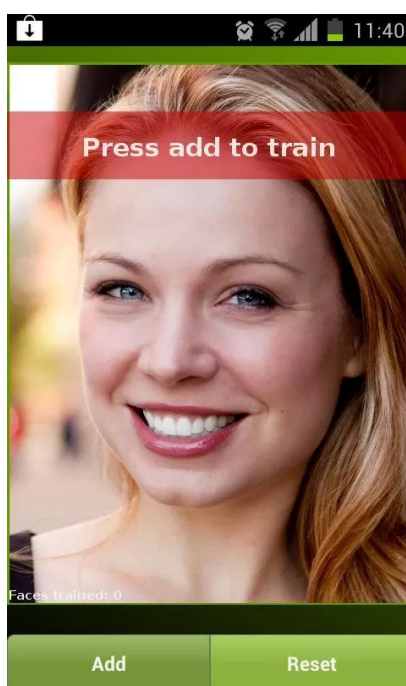


Figura 8 – AppLock – Reconhecimento de faces

Desenvolvida sobre a plataforma VDFaceSDK da Visidon - um conjunto de bibliotecas para construir soluções de reconhecimento de faces – reutiliza grande parte das funcionalidades de reconhecimento, detecção e verificação de faces, assim como detecção de movimento do utilizador ao tirar fotografias (sorriso, piscar de olhos) [Visidon, 2013b].

Publicitado pela empresa como a aplicação de maior sucesso para dispositivos móveis no ramo do reconhecimento de faces, o AppLock conta já com mais de quinhentas mil instalações a partir do Google Play, recebendo *feedback* positivo dos seus utilizadores [Google, 2012].

4.3 Tecnologias de reconhecimento em dispositivos móveis

As tecnologias de reconhecimento estão tradicionalmente associadas a computadores *desktop* ou servidores, uma vez que necessitam de grandes quantidades de processamento para obterem resultados fidedignos. No entanto, com o avanço das tecnologias móveis, e o crescimento do poder de computação dos dispositivos, a migração das bibliotecas tradicionais de computação gráfica começa a surgir. Bibliotecas como OpenCV, FastCV ou Camellia são o resultado das adaptações de plataformas e técnicas tradicionais para processadores de dispositivos móveis. Nesta secção irão ser apresentadas algumas ferramentas para a implementação de módulos de reconhecimento de imagem em sistemas operativos móveis.

4.3.1 OpenCV

Open Source Computer Vision (OpenCV) é, como o nome indica, uma biblioteca de código aberto para processamento gráfico com uma componente de aprendizagem de sistemas. Com interfaces para um vasto conjunto de linguagens e bibliotecas de programação, o OpenCV é compatível com as plataformas *desktop* Windows, Linux e Mac, e com as plataformas móveis Android e iOS [Itseez, 2013a]. Construído com o intuito de servir de infraestrutura comum a inúmeras aplicações relacionadas com robótica e computação gráfica, e sempre com o objetivo de atingir a máxima eficiência computacional, esta biblioteca acelera o desenvolvimento de produtos comerciais capazes de tirar o máximo partido do multiprocessamento em tempo real dos dispositivos atuais.

Escrita originalmente em C/C++, disponibiliza camadas de tradução das funções básicas para outras linguagens de programação como Python e Java, atingindo um universo de quarenta e sete mil pessoas ativas na comunidade do OpenCV, com o total de descargas da plataforma a exceder os sete milhões [Itseez, 2013b].

Com mais de dois mil e quinhentos algoritmos otimizados para servir áreas como arte interativa, inspeção de minas, informação geográfica ou robótica [Itseez, 2013b], esta plataforma implementa tanto técnicas clássicas de computação gráfica e aprendizagem de sistemas, como as soluções mais recentes propostas pela comunidade científica. Neste vasto conjunto podem ser encontrados algoritmos para detetar e reconhecer faces, identificar objetos, classificar ações humanas em vídeos, agregar imagens para produzir uma única imagem de alta resolução, encontrar imagens semelhantes numa base de dados, introduzir realidade aumentada numa cena de objetos, entre outros.

Distribuída sobre os termos da licença BSD, esta biblioteca é de livre utilização tanto para uso comercial como académico, permitindo que entidades cedam e alterem o código conforme necessário.

4.3.2 FastCV

O projeto FastCV é o resultado do esforço realizado pela empresa Qualcomm na construção de uma biblioteca e um conjunto de ferramentas de desenvolvimento para processamento de imagem, destinados exclusivamente para suportar o desenvolvimento de aplicações com capacidades gráficas avançadas em dispositivos móveis. Desenhada especificamente para ter uma performance exímia no sistema operativo *Android*, esta biblioteca, ainda na sua versão 1.0, é distribuída sobre a forma de um único ficheiro binário, contendo código otimizado para processadores ARM e Snapdragon [Qualcomm, 2011b].

Apresentada como sendo a primeira biblioteca de processamento de gráficos e imagem dedicada para as arquiteturas dos dispositivos móveis, a biblioteca FastCV disponibiliza interfaces de programação com capacidades de aceleração por *hardware*, especialmente em processadores Snapdragon desenvolvidos pela empresa [Qualcomm, 2011a]. Devido à sua vasta coleção de funções para processamento de imagem a baixo nível e interação com sensores e câmaras, este projeto constitui a base da solução da Qualcomm para realidade aumentada: *Vuforia Augmented Reality*. Segundo a empresa, o FastCV pode ser utilizado diretamente nas aplicações, ou utilizado por programadores de *middleware* no desenho e conceção de plataformas de reconhecimento de gestos, faces, e objetos, assim como a deteção de movimento, cálculo de profundidade de campo, reconhecimento de caracteres, entre outros [Qualcomm, 2011a].

Assente numa arquitetura modular, o FastCV é composto por diversos componentes como operações matemáticas e vetores, processamento de imagem, transformação de imagem, deteção de áreas de interesse, deteção de objetos, reconstrução 3D, *clustering* e pesquisa e seguimento de objetos e movimento, mantendo uma documentação extensa sobre cada um destes módulos [Qualcomm, 2011c].

Com o conjunto de ferramentas de desenvolvimento do FastCV *Android* disponível para Windows, Mac e Linux sobre a licença FastCV SDK License (sem custos de utilização), a equipa da Qualcomm responsável pelo projeto prevê o lançamento desta solução para outros sistemas operativos móveis como iOS e Windows Phone [Qualcomm, 2011b].

4.3.3 Camellia Library

Camellia é o resultado de uma colaboração entre parceiros como École des Mines de Paris (ENSMP), Philips, Universidade de Hannover, Universidade de Las Palmas e Renault, que se reuniram para o desenvolvimento de uma biblioteca de processamento de gráficos e imagem, com o intuito de ser multiplataforma, robusta e com a máxima performance. Totalmente escrita em C, esta biblioteca de código livre inclui funções para o processamento de imagem convencional como aplicar filtros nas imagens, distorção, manipulação de gráficos, e conversão de cores [Camellia, 2013].

Encarado como o projeto substituto do Intel IPL, atualmente descontinuado, o Camellia mantém completa interoperabilidade com o OpenCV, embora adicionando camadas de tradução para facilitar a compreensão das funções. Possível de ser utilizado na linguagem C++, permitindo às equipas de desenvolvimento retirar todas as vantagens da programação orientada a objetos, assim como uma interface para a linguagem interpretada Ruby, que inclui o suporte a exceções e otimização de memória [Camellia, 2013].

Suportando imagens até 16 bits de profundidade, processamento de regiões de interesse, e aplicação de máscaras de bits, contém algoritmos exclusivos para processamento de imagem como Watershed hierárquico e estimativa do movimento 3DRS com recurso a RLE (Run-Length Encoded) para efetuar cálculos de iluminação mais eficientemente [Camellia, 2013].

Distribuído sobre a licença BSD e publicado através do sítio de alojamento de *software* Sourceforge, este projeto pretende crescer com a comunidade apelando a contribuições para manter a biblioteca atualizada e proeminente na área do processamento de imagem [Camellia, 2013].

4.3.4 Neuroph

Neuroph é o nome de uma plataforma Java para conceber arquiteturas aplicacionais com base em redes neuronais. Criado através de um projeto final de licenciatura, e evoluído no âmbito de uma tese de mestrado, foi lançado como projeto de código aberto na plataforma Sourceforge em Setembro de 2008 [Neuroph, 2013c]. Desenhado para constituir uma plataforma minimalista de redes neuronais, o Neuroph é pequeno, intuitivo e de fácil utilização e aprendizagem, além de flexível à inclusão de módulos e extensões, apesar de todas as funcionalidades já prontas a utilizar [Neuroph, 2013b].

A extensa documentação aliada às ferramentas que agilizam as tarefas comuns no desenvolvimento de redes neuronais nas fases de criação, treino, teste e distribuição, tornam a plataforma numa ótima alternativa para a implementação deste tipo de sistemas [Neuroph, 2013b].

Com o editor gráfico de redes neuronais incluído no Neuroph, até o utilizador mais inexperiente é capaz de iniciar a aprendizagem nesta área, compreendendo facilmente os conceitos básicos inerentes à estruturação de uma rede neuronal simples [Neuroph, 2013a]. As redes criadas com o Neuroph podem ser utilizadas para classificação, reconhecimento, previsão e modelação, permitindo gerar aplicações para reconhecimento de imagem, previsão de *stocks*, classificação de padrões, entre outras [Neuroph, 2013c]. Atualmente, a plataforma implementa redes neuronais do tipo Adaline Perceptron, Hopfield, Maxnet, Neru Fuzzy Reasoner, entre outros, suportando tanto métodos de classificação supervisionados ou não supervisionados [Neuroph, 2013a].

Disponível sobre a licença Apache 2.0, é de utilização livre por empresas e particulares para fins comerciais ou não comerciais [Neuroph, 2013a].

4.3.5 Moodstocks

A abordagem tomada pela empresa Moodstocks para o reconhecimento de imagem dista das restantes pela arquitetura orientada à disponibilização de um serviço, em que todas as aplicações cliente consultam servidores dedicados da empresa para efetuar o processamento das imagens [Moodstocks, 2013a]. Com o foco no suporte a aplicações móveis, esta plataforma é escalável a vários milhares de imagens por aplicação, flexível, e capaz de retornar resultados precisos em poucos milissegundos [Moodstocks, 2013a].

Apesar da dependência dos servidores da Moodstocks numa primeira instância para preparação das imagens de treino, todo o processo de reconhecimento de novas imagens é efetuado nos dispositivos móveis, consultando bases de dados locais com imagens pré-processadas [Moodstocks, 2013a].

Após efetuar o reconhecimento de uma imagem, o utilizador tem ao seu dispor um conjunto de ações passíveis de serem executadas com base nas informações recolhidas sobre a fotografia [Moodstocks, 2013a]. Tolerante a ruído e desfoque nas imagens, este sistema possui uma aplicação de sincronização das bases de dados locais com os servidores Moodstocks, permitindo assim ao programador adicionar novas imagens ao conjunto de treino, aumentando a sua fiabilidade [Moodstocks, 2013a].

Para ter acesso à plataforma, é necessário efetuar um registo na Moodstocks e selecionar um plano de serviços, que vão desde planos sem custos (para pequenas aplicações) até planos empresariais. Segundo a empresa, os exemplos disponibilizados permitem gerar aplicações com capacidades de reconhecimento de imagem sem codificar uma única linha de código, mantendo simultaneamente a aplicação universal, robusta, leve e multi-plataforma [Moodstocks, 2013b].

4.3.6 Kooaba

Kooaba apresenta-se como a plataforma líder na área de reconhecimento de imagem baseada na nuvem⁵, fornecendo um serviço de elevado desempenho aos seus clientes. Com uma base de dados de mais de sessenta milhões de imagens, esta plataforma possibilita um reconhecimento ímpar para dispositivos móveis, estando em constante atualização das suas bases de dados e algoritmos [Kooaba, 2011a].

Com um ótimo reconhecimento de caras, texto, monumentos, produtos e logótipos, o Kooaba conta já com aplicações desenvolvidas para várias áreas da indústria, como a indústria vinícola, cinema, compras, turismo, entre outras [Kooaba, 2011a].

Suportando bases de dados públicas e privadas, com possibilidade de associar várias imagens a um único tipo de objeto (para reconhecimento em 3D), definição de novos tipos de objetos, estatísticas em tempo real e suporte em todos os momentos, torna-se um sistema apelativo a empresas e individuais para a implementação dos componentes de reconhecimento das aplicações. Numa arquitetura completamente baseada na nuvem, o reconhecimento de imagem é feito através de consultas aos servidores da Kooaba, sendo necessário um registo prévio para acesso à API [Kooaba, 2011a].

Com vários planos para as mais diversas necessidades dos clientes, o Kooaba é de livre utilização para aplicações até um número máximo estabelecido de pedidos de reconhecimento por mês, carecendo de pagamento se esse limite for ultrapassado [Kooaba, 2011b].

⁵ Nuvem – processamento levado a cabo por um conjunto de computadores interligados por uma rede de comunicação de tempo real.

4.4 Análise comparativa das tecnologias de reconhecimento analisadas

Os projetos analisados neste capítulo retratam a oferta atual do mercado no que diz respeito a tecnologias de reconhecimento de imagem, especialmente focadas em dispositivos móveis. Com o levantamento e análise efetuado neste capítulo, foi possível identificar inúmeras bibliotecas que, apesar de possibilitarem o reconhecimento de imagem, são apenas camadas de tradução de funções de outras bibliotecas, não trazendo valor acrescentado na sua escolha. O quadro que se segue (ver Tabela 3) realça as principais diferenças encontradas entre os sistemas e resume as suas principais características, permitindo posteriormente optar pela tecnologia a integrar no projeto.

Tabela 3 — Tecnologias de reconhecimento - Comparação das soluções encontradas

	OpenCV	FastCV	Camellia	Neuroph	MoodStocks	Kooba
Deteção de objetos em imagem?	Sim	Sim	Sim, através de OpenCV	Sim	Sim	Sim
Classificação de objetos em imagem?	Sim	Sim	Sim, através de OpenCV	Sim	Sim	Sim
Processamento de imagens <i>offline</i>?	Sim	Sim	Sim	Sim	Sim, após pré-processamento <i>online</i>	Não
Suporte à aprendizagem do sistema?	Sim	Não	Sim, através de OpenCV	Sim	Sim	Sim
Tipo de licença?	Licença BSD (livre)	Licença FastCV SDK (livre)	Licença BSD (livre)	Licença Apache 2.0 (livre)	Subscrição de um plano	Subscrição de um plano

- “Deteção de objetos em imagem?”: indica se a plataforma é capaz de detetar objetos numa imagem;
- “Classificação de objetos em imagem?”: refere se a alternativa permite a atribuição de classes a tipos de objetos;
- “Processamento de imagens *offline*?”: indica se a solução suporta o processamento das imagens no dispositivo móvel (sem acesso a serviços exteriores);
- “Suporte à aprendizagem do sistema?”: refere se a plataforma possibilita a aprendizagem através da inclusão de cada vez mais objetos;

- “Tipo de licença?”: descreve o tipo de licenciamento a que cada uma das soluções está sujeita.

Os parâmetros utilizados nesta comparação representam o conjunto de características ideais a que as plataformas necessitariam de corresponder para a implementação com sucesso de um sistema de reconhecimento de imagem. As restrições orçamentais do projeto obrigaram, tal como efetuado em outras análises desta investigação, à inclusão do tipo de licenciamento praticado em cada uma das soluções.

4.5 Conclusões

O levantamento do estado da arte efetuado neste capítulo suportou, de uma forma sustentada, a tomada de decisão da equipa do projeto por uma tecnologia de reconhecimento de imagem a ser integrada na plataforma a desenvolver. A análise minuciosa de cada uma das alternativas e a ponderação das suas vantagens e desvantagens permitiu justificar a opção tomada, e simultaneamente, descobrir funcionalidades passíveis de incluir no sistema a desenvolver, assim como definir os limites da sua implementação.

Numa primeira instância, foram estudadas as aplicações existentes no mercado móvel que implementam com sucesso algoritmos de reconhecimento em imagem, averiguando o seu modo de operação e filosofia de distribuição aplicada aos produtos. Depois de identificadas as várias vertentes de negócio na área de reconhecimento de imagem móvel, foram estudadas duas aplicações com abordagens distintas, que resumem a oferta disponibilizada pelo mercado nesta área: Google Goggles e Visidon AppLock. Como estas soluções não revelam as tecnologias que utilizam nem divulgam quaisquer informações sobre a sua implementação, foi necessário partir para a identificação e triagem das tecnologias mais proeminentes para reconhecimento de imagem em dispositivos móveis. Feita a análise detalhada de cada uma das alternativas encontradas, foi construído um quadro de comparação que confronta todas as soluções com um conjunto pré-determinado de parâmetros, a fim de apurar a tecnologia mais adequada para integração no projeto. O resultado da comparação comprovou claramente a supremacia da tecnologia OpenCV, muito utilizada no ramo da robótica, processamento de gráficos e especialmente em reconhecimento de padrões e imagem. Livre para utilização com fins comerciais, esta biblioteca permite a deteção, classificação e consequente reconhecimento de imagens em modo *offline*, e ainda o suporte a um módulo para aprendizagem automática de novas imagens. Devido ao desenvolvimento contínuo ao longo do tempo, e a camadas de tradução da API para as linguagens de programação mais comuns, existe uma comunidade forte no apoio e documentação de toda a plataforma, que alicia ao seu uso.

Assim sendo, a escolha de uma tecnologia de reconhecimento de imagem recai sobre o OpenCV, que será integrado na aplicação a desenvolver e fará parte da versão final da solução.

5 Implementação e Avaliação

5.1 Introdução

A revolução ocorrida no mundo tecnológico devido ao aparecimento da computação móvel incitou o aparecimento de novos paradigmas de interação entre pessoas e sistemas, nomeadamente a interação com dispositivos móveis como *smartphones* ou *tablets*. A facilidade de uso, utilidade e satisfação associadas a estes dispositivos, tornam a computação ubíqua uma realidade evidente, através de aplicações baseadas em sensores precisos e conscientes do contexto envolvente. Desta emergência do mercado móvel, surgiram inúmeras áreas de negócio capazes de explorar o seu potencial, como o desenvolvimento de aplicações de foro cívico e serviços orientados ao cidadão.

A notificação de problemas urbanos constitui uma excelente oportunidade de negócio, não só para as entidades municipais, que conseguem assim diminuir os seus custos de operação, mas também para os cidadãos, que assumem um papel ativo na sua comunidade local, promovendo o seu crescimento e prosperidade. O retorno de investimento do mercado global das aplicações móveis justifica a investigação e desenvolvimentos já efetuados nesta área, pelo que soluções para a notificação de problemas urbanos já não são novidade. A existência de propostas de *standards* de relatórios de ocorrências, assim como a disponibilização de serviços universais de consulta e notificação de problemas, geram uma alta competitividade neste ramo, levando ao aparecimento de alternativas aos sistemas atuais e à reinvenção de funcionalidades. No entanto, a adaptação deste tipo de sistemas às constantes alterações da realidade tecnológica é feita gradualmente, deixando margem para o aparecimento de soluções concorrentes. No estudo efetuado foram identificadas falhas nos sistemas atuais de notificação de problemas, como a falta de identificação clara da patologia a reportar, a complexidade associada à submissão de novas ocorrências ou a ausência de um mecanismo de discussão dos problemas.

As secções seguintes pretendem descrever em detalhe todo o processo de desenvolvimento de uma solução para a notificação de problemas urbanos que, para além das funcionalidades

típicas associadas a este tipo de aplicações, inova na forma de submissão de novas ocorrências, quer através de mecanismos de demarcação manual das zonas problemáticas numa imagem, como em algoritmos para deteção, reconhecimento e classificação das situações reportadas. Em primeiro lugar, é apresentado o sistema de informação geográfico de base de toda a solução, em que a aplicação se irá inserir. De seguida, é feita uma proposta de um modelo para a sua implementação, onde são capturados os requisitos funcionais e não funcionais, concebida a arquitetura da solução e desenhados os módulos específicos para a anotação e reconhecimento de imagens. Posteriormente são indicados todos os passos seguidos para a implementação da solução, e feita uma avaliação da eficácia dos desenvolvimentos efetuados. Por fim são extraídas conclusões do trabalho desenvolvido.

5.2 Sistema de informação geográfico

Um sistema de informação geográfico ou infraestrutura de dados espaciais, representa uma ferramenta fundamental no apoio à tomada de decisão por parte das entidades municipais no planeamento e desenvolvimento regional [Oliveira et al., 2012]. Com a necessidade crescente de os municípios utilizarem a informação de base territorial de forma a responderem a necessidades específicas das comunidades locais, os SIG (Sistemas de Informação Geográfica) implementam normas internacionais para garantir a interoperabilidade da informação a processar [Oliveira et al., 2012].

Nesta secção irá ser apresentado o sistema de informação geográfica que serve de base para o alojamento da aplicação setorial a desenvolver.

5.2.1 Portal SIG

Idealizado como um portal com capacidades geográficas, este sistema permite a gestão da informação territorial de um município de forma integrada, sendo possível representar os vários níveis de informação de uma região como ecopontos, pontos de água, hotéis, ciclovias, etc. Além das funcionalidades de base geográfica, o portal inclui módulos para gestão de utilizadores, grupos de utilizadores, níveis de informação e categorias, além do suporte à adição de aplicações setoriais independentes (ver Figura 9).

5.2.1.1 Arquitetura do portal

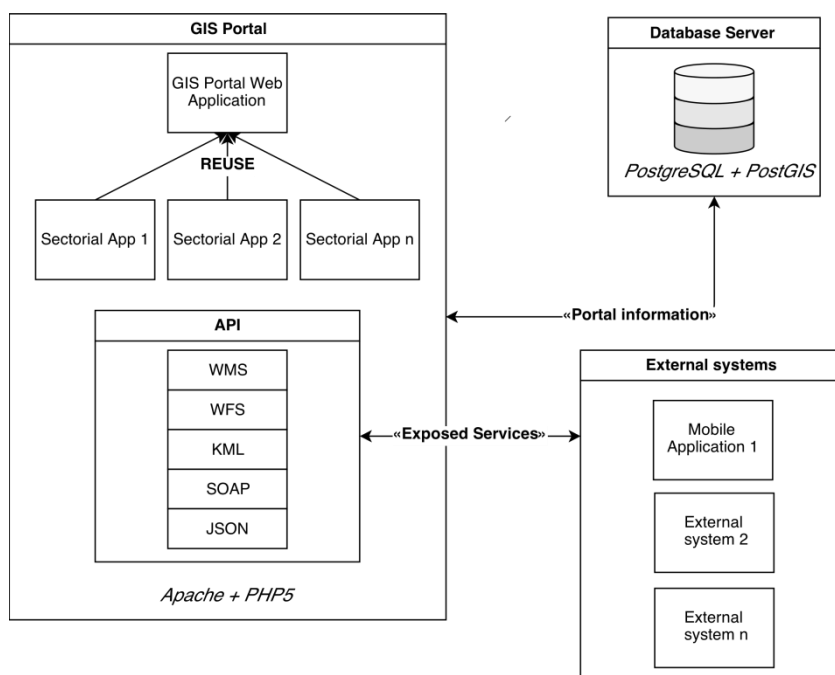


Figura 9 – GeoAve – Arquitetura do portal

O portal descrito nesta secção, desenvolvido na íntegra com base em tecnologias de código livre, foi concebido com o intuito de ser modular, interoperável e de simples utilização, permitindo agregar um conjunto de aplicações setoriais que se encarregam de um domínio específico.

A arquitetura da solução prevê a independência da informação das aplicações setoriais, apesar de lhes facilitar o acesso à informação global do portal como os utilizadores, informação territorial e níveis de informação.

A interoperabilidade entre sistemas é atingida pela implementação dos *standards* definidos pela comunidade europeia, como o WMS ou WFS, permitindo a sistemas externos utilizar a informação disponibilizada pelo portal para consulta a qualquer altura. O suporte destes e outros padrões como JSON e SOAP, possibilita o desenvolvimento de aplicações externas que consomem a informação existente no portal e, recorrendo a serviços implementados no portal ou aplicações setoriais, atualizam essa mesma informação.

5.2.1.2 Mapa

Numa lógica de abstração da informação a representar, é possível incluir no portal elementos georreferenciados como pontos, linhas e polígonos, assim como obter informação sobre esses elementos de diversas fontes externas, tendo por base a sua localização geográfica.

O Mapa, encarado como uma aplicação pertencente ao núcleo do sistema, implementa grande parte da lógica de apresentação e gestão de níveis de informação do portal. Com o objetivo de maximizar a área útil do ecrã para navegação no mapa, a obtenção dos pontos de interesse associados a categorias e níveis de informação é feita com recurso a uma árvore de navegação, sendo possível de esconder em qualquer momento (ver Figura 10).

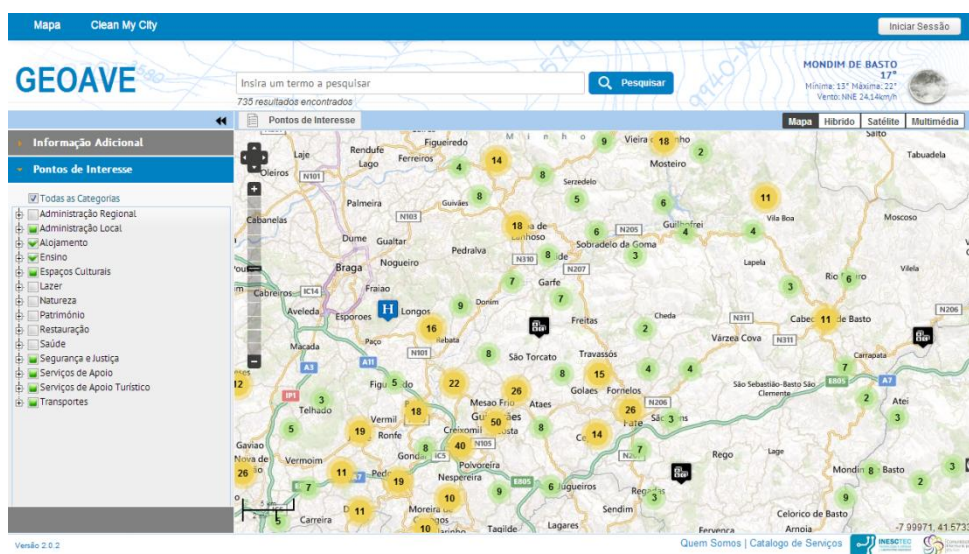


Figura 10 – GeoAve – Mapa

5.2.1.3 Administração

Através de um painel para gerir a informação pública constante no mapa, utilizadores com privilégios de administração podem adicionar categorias e subcategorias de informação, atributos, simbologias e pontos de interesse através de uma interface simples e amigável (ver Figura 11).

ID	Nome do Estilo	Ícone Associado		
11	Municípios		Substituir ícone	Apagar
12	Freguesias		Substituir ícone	Apagar
13	Hotel		Substituir ícone	Apagar
14	Parques de Campismo		Substituir ícone	Apagar
15	Agroturismo		Substituir ícone	Apagar
16	Turismo de Habitação		Substituir ícone	Apagar
17	Básico e secundário		Substituir ícone	Apagar

Figura 11 – GeoAve – Administração do Mapa

Para além da administração da aplicação Mapa, o portal inclui uma administração geral encarregada de gerir toda a informação diretamente relacionada com o portal, como utilizadores, grupos, aplicações setoriais e permissões associadas aos menus dessas aplicações.

5.2.2 Aplicações Setoriais

A informação territorial de base do portal fornece uma visão genérica dos vários níveis de gestão municipal existentes numa região, possibilitando a consulta de informação detalhada sobre os vários pontos de interesse inseridos no sistema. No entanto, a necessidade de otimização de processos e o aparecimento de novas formas de interação com a comunidade local, conduziram à conceção de uma arquitetura que sustentasse a adição de aplicações e serviços com um fim específico, sendo igualmente sensíveis ao território e orientados ao tratamento especializado de um domínio de gestão municipal concreto [Oliveira et al., 2012]. Áreas da gestão local como a recolha de lixo, viaturas abandonadas ou problemas de saneamento, são apenas alguns exemplos de possíveis aplicações setoriais a considerar no contexto do sistema de informação geográfico. Estas instâncias, apesar de integradas com o portal, são independentes deste na sua génese, podendo conter recursos e informação próprios [Oliveira et al., 2012].

5.2.2.1 CleanMyCity

O CleanMyCity é um caso de sucesso na implementação e integração de uma aplicação setorial no portal SIG, reutilizando os seus conceitos e serviços de base geográfica e alargando as possibilidades de manuseamento da informação regional relacionada com a recolha de resíduos. Numa abordagem orientada à disponibilização de um serviço público para o cidadão, e simultaneamente numa tentativa de otimização dos processos associados à recolha de resíduos dos municípios, o CleanMyCity possibilita aos cidadãos de uma comunidade visualizarem dados reais da localização e disponibilidade de contentores de resíduos, roupa e indiferenciados, com vista à determinação do contentor mais próximo da sua posição (ver Figura 12).

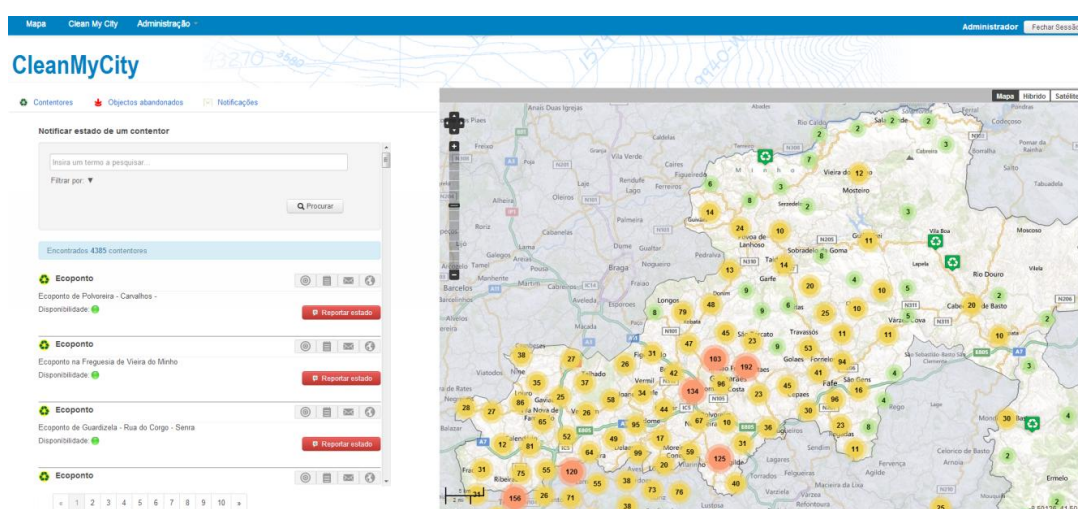


Figura 12 – GeoAve – CleanMyCity

herdando as características de sistemas de *crowd-sourcing*, esta aplicação confia nos seus cidadãos a tarefa de manter atualizada a informação da disponibilidade dos vários contentores, permitindo que reportem o estado de um contentor como cheio ou danificado. Outra das funcionalidades de reaproveitamento do esforço dos utilizadores é a notificação de objetos abandonados na via pública. Através de um formulário simples, os utilizadores são incitados a reportar um objeto abandonado na sua área de residência, descrevendo a ocorrência e opcionalmente anexando uma imagem. O seguimento quer das ocorrências de objetos abandonados, quer de relatórios de disponibilidade de contentores é feito pela aplicação de Administração do CleanMyCity, direcionada para os funcionários das entidades de gestão local. Através desta interface, os funcionários responsáveis pelo encaminhamento e resolução das notificações processam as mensagens enviadas pelos utilizadores, atualizam a disponibilidade dos contentores e enviam mensagens gerais para a comunidade.

Para ir de encontro às necessidades dos seus utilizadores, e assim atingir o maior número de cidadãos, o CleanMyCity está também disponível numa vertente móvel *Android*, em que os utilizadores recebem a informação de contentores e mensagens de objetos abandonados filtrados com base na sua localização geográfica.

5.3 Modelo proposto

O modelo proposto nesta secção pretende dar uma resposta efetiva às necessidades identificadas no âmbito deste projeto de investigação. A oportunidade de negócio criada no mercado dos sistemas do foro cívico conduziu à idealização, conceção e implementação de uma solução para a notificação de problemas urbanos orientado ao cidadão e ao ato de exercer cidadania.

A experiência detida pela equipa na área dos sistemas de informação geográficos, aliada à recente necessidade da implementação de serviços municipais especializados, reuniram as condições ideais para a concretização do projeto. Reaproveitando toda a infraestrutura de base do portal SIG, e seguindo a lógica de reutilização das suas camadas, a solução para a notificação de problemas urbanos passa pelo desenvolvimento de uma aplicação setorial e serviços associados. O sistema abarcará várias vertentes:

- uma vertente *Web*, composta por uma aplicação para o público em geral reportar incidentes, e uma área de administração para que as entidades municipais possam dar o acompanhamento devido às ocorrências submetidas;
- uma vertente orientada à disponibilização de serviços, implementada no portal SIG que servirá de ponto de acesso à realização de operações na base de dados do sistema;
- uma vertente móvel para possibilitar aos cidadãos submeter notificações a partir de *smartphones* ou *tablets*.

As seguintes secções procuram documentar o processo associado à engenharia de requisitos do sistema a desenvolver, e a consequente especificação formal da solução.

Numa primeira fase, são delineados os casos de uso do sistema e os atores intervenientes, documentando em formato breve cada uma dessas situações de utilização que constituem os requisitos funcionais da aplicação. São ainda capturados os requisitos não funcionais da plataforma numa especificação suplementar, que responde a questões como usabilidade, confiabilidade, desempenho e manutenção.

Numa fase posterior ao levantamento de requisitos, é apresentada a arquitetura concebida para suportar a solução de notificação de problemas urbanos, detalhando as funções de cada um dos componentes intervenientes no sistema e a sua forma de interação/comunicação. Feito o enquadramento da arquitetura global do sistema, segue-se a descrição pormenorizada dos módulos mais relevantes da solução, como a anotação e o reconhecimento de objetos em imagem.

5.3.1 Requisitos funcionais e não funcionais

5.3.1.1 Atores do sistema

O levantamento dos requisitos para a delineação da solução envolve, em primeiro lugar, a correta identificação dos atores e partes interessadas (*stakeholders*) intervenientes no sistema. No âmbito deste projeto, foram reconhecidos os seguintes atores:

- **Cidadão:** indivíduo com um papel ativo na sua comunidade local, encarregado de interagir com as várias aplicações do sistema para consultar, reportar e discutir ocorrências;
- **Funcionário municipal:** indivíduo pertencente a uma entidade de gestão governamental local, responsável pelo encaminhamento e resolução das ocorrências reportadas;
- **Entidades externas/serviços:** indivíduo ou sistema capaz de consumir serviços para consulta, notificação e discussão de ocorrências.

Estes atores correspondem à totalidade de intervenientes no sistema como um todo, englobando as aplicações das vertentes móvel, *Web* e camada de serviços.

5.3.1.2 Casos de uso

Nesta secção irão ser apresentados dois diagramas de casos de uso, um que ilustra as interações entre os atores e o sistema como um todo (encarado como o diagrama de casos de uso de nível zero, ver Figura 13) e um diagrama focado nas funcionalidades disponibilizadas pela aplicação móvel (ver Figura 14). Devido à complexidade do sistema em causa, ainda só existe documentação formal para a componente móvel, o que obriga à limitação do âmbito da especificação.

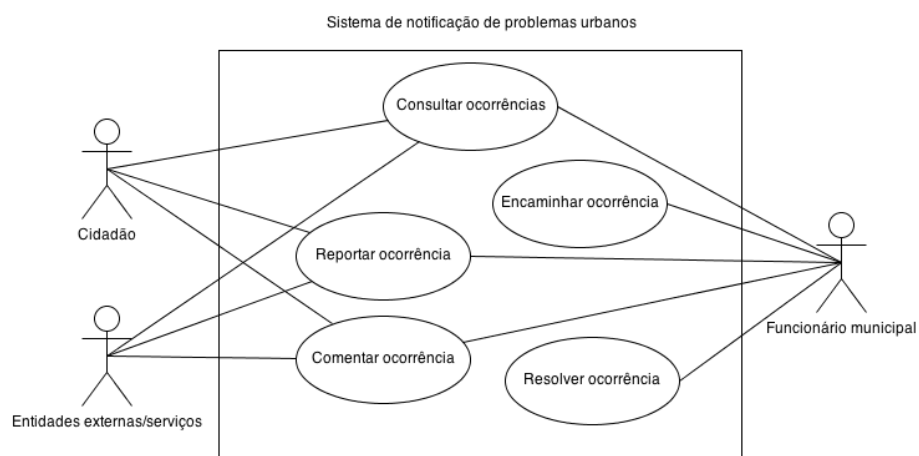


Figura 13 – Diagrama de casos de uso – Sistema (nível zero)

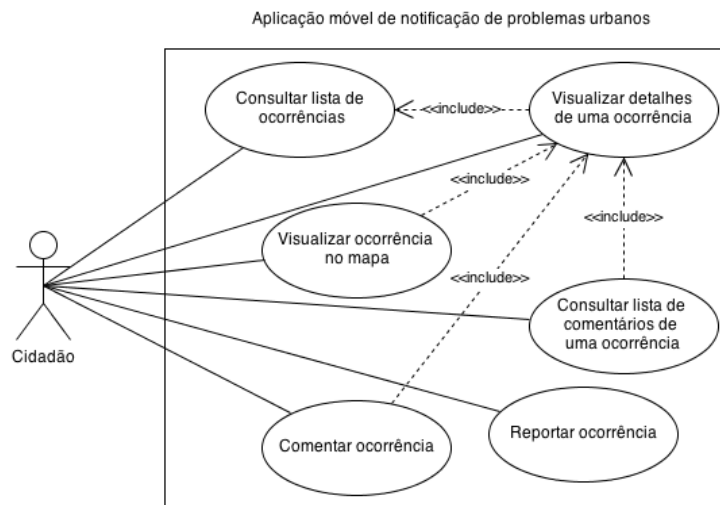


Figura 14 – Diagrama de casos de uso – Aplicação móvel

Esclarecida a comunicação entre atores e sistema com recurso aos diagramas, os casos de uso são descritos formalmente seguindo o formato breve:

Caso de uso: UC1

Nome: Consultar ocorrências

Atores: Cidadão, Entidades externas/serviços, Funcionário municipal

Descrição (fluxo principal): O utilizador acede ao sistema e requisita a listagem de ocorrências de acordo com um conjunto de parâmetros pré-estabelecidos (raio em torno de localização, data da notificação, etc.). O sistema retorna a lista de ocorrências filtrada pelos parâmetros indicados.

Caso de uso: UC2

Nome: Reportar ocorrência

Atores: Cidadão, Entidades externas/serviços, Funcionário municipal

Descrição (fluxo principal): O utilizador indica ao sistema a intenção de reportar uma ocorrência na plataforma. O sistema apresenta um formulário ao utilizador para preenchimento das informações da nova ocorrência, como título, descrição, anexos (imagens, vídeo ou áudio), localização e *tags* para classificar a ocorrência. O utilizador preenche os dados necessários e submete o formulário. O sistema informa do sucesso ou insucesso da operação e apresenta a lista de ocorrências atualizada.

Caso de uso: UC3**Nome:** Comentar ocorrência**Atores:** Cidadão, Entidades externas/serviços, Funcionário municipal

Descrição (fluxo principal): A partir do ecrã de detalhe de uma ocorrência, o utilizador indica que deseja adicionar um comentário à discussão atual. O sistema apresenta um formulário para preenchimento do texto e possíveis anexos a submeter. Após o preenchimento, o utilizador submete o formulário. O sistema informa do sucesso ou insucesso da operação, e apresenta a lista de comentários atualizada.

Caso de uso: UC4**Nome:** Encaminhar ocorrência**Atores:** Funcionário municipal

Descrição (fluxo principal): O Funcionário municipal seleciona da lista de ocorrências a que deseja encaminhar para as entidades responsáveis. Após selecionar a ocorrência, e indicar que a pretende encaminhar, o sistema apresenta uma lista com as opções possíveis para o encaminhamento. Feita a escolha da entidade responsável pela resolução da ocorrência, o utilizador confirma todo o procedimento e submete o formulário. O sistema informa do sucesso ou insucesso da operação, e apresenta a lista de ocorrências por encaminhar ao utilizador.

Caso de uso: UC5**Nome:** Resolver ocorrência**Atores:** Funcionário municipal

Descrição (fluxo principal): O Funcionário municipal seleciona da lista de ocorrências a que pretende marcar como resolvida pelas entidades responsáveis. Após selecionar a ocorrência, e indicar que a pretende resolver, o sistema apresenta uma lista com as entidades possíveis de terem efetuado a resolução ou, caso tenha sido indicada a entidade num processo de encaminhamento, o sistema sugere essa entidade para a resolução. Feita a escolha da entidade responsável pela resolução da ocorrência, o utilizador confirma todo o procedimento e submete o formulário. O sistema informa do sucesso ou insucesso da operação, e apresenta a lista de ocorrências por resolver ao utilizador.

Caso de uso: UC6

Nome: Consultar lista de ocorrências

Atores: Cidadão

Descrição (fluxo principal): O cidadão acede à aplicação e requisita a listagem de ocorrências correspondente a um filtro pré-estabelecido (ocorrências mais próximas do utilizador, ocorrências mais recentes, ocorrências mais populares e ocorrências mais comentadas). O sistema retorna a lista de ocorrências filtrada pelos parâmetros indicados.

Caso de uso: UC7

Nome: Visualizar detalhes de uma ocorrência

Atores: Cidadão

Descrição (fluxo principal): Incluir obrigatoriamente os passos do caso de uso UC6. O utilizador seleciona uma ocorrência da lista e indica ao sistema que pretende visualizar os detalhes. O sistema apresenta o ecrã de informações sobre a ocorrência.

Caso de uso: UC8

Nome: Consultar lista de comentários de uma ocorrência

Atores: Cidadão

Descrição (fluxo principal): Incluir obrigatoriamente os passos do caso de uso UC6. O utilizador seleciona uma ocorrência da lista e indica ao sistema que pretende consultar a lista de comentários de uma ocorrência. O sistema apresenta o ecrã de comentários relativos à ocorrência.

Caso de uso: UC9

Nome: Visualizar ocorrência no mapa

Atores: Cidadão

Descrição (fluxo principal): Incluir obrigatoriamente os passos do caso de uso UC6. O utilizador seleciona uma ocorrência da lista e indica ao sistema que pretende visualizar no

mapa. O sistema apresenta o ecrã com o mapa e coloca um marcador na localização da ocorrência.

Caso de uso: UC10

Nome: Comentar ocorrência

Atores: Cidadão

Descrição (fluxo principal): Incluir obrigatoriamente os passos do caso de uso UC8. O utilizador indica ao sistema que pretende adicionar um comentário à discussão. O sistema apresenta um formulário para preenchimento do texto e possíveis anexos a submeter. Após o preenchimento, o utilizador submete o formulário. O sistema informa do sucesso ou insucesso da operação, e apresenta a lista de comentários atualizada.

De notar a repetição de alguns casos de uso com o mesmo propósito, como o UC1 e UC6 que, devido às adaptações necessárias para a realidade das plataformas móveis, sofreram alterações relevantes no seu fluxo. Apesar do resultado do caso de uso em ambos ser idêntico, o modo de funcionamento foi alterado para corresponder às expectativas dos utilizadores de dispositivos móveis.

5.3.1.3 Especificação suplementar

Para ser possível capturar os requisitos não funcionais de forma consciente, estruturada e persistente, foi construída uma especificação suplementar⁶ do sistema a desenvolver. Esta especificação procura documentar o conjunto de fatores que influenciam a arquitetura da solução e que condicionam a sua implementação, como a funcionalidade, usabilidade, confiabilidade, desempenho e manutenção.

Funcionalidade

- Autenticação na plataforma: para poder identificar os utilizadores do sistema na submissão de notificações, comentários e votos;
- Salva-guarda da informação: toda a informação que o utilizador pretende enviar para o sistema deve ser copiada e mantida em bases de dados locais para evitar a sua perda.

⁶ Artefacto para documentação dos requisitos não funcionais. (formato retirado de: <http://www.sis.pitt.edu/~srinivas/isad/lecture07/egss.htm>)

Usabilidade

- A interface do sistema deve-se adaptar a vários tipos de ecrã e dispositivos de entrada;
- As interações com o utilizador devem ser breves e objetivas, requerendo o mínimo indispensável de informação;
- As operações mais complexas no sistema devem ter documentação de ajuda associada.

Confiabilidade

- Disponibilidade: o sistema deve estar disponível para uso em qualquer altura, sendo tolerável uma taxa de 5% de inatividade da plataforma;
- Segurança: o acesso aos servidores da plataforma deve ser restrito apenas a administradores bem identificados;
- Integridade: As submissões efetuadas pelos utilizadores devem desencadear operações transacionais nas bases de dados, deixando sempre o sistema num estado íntegro.

Desempenho

- Múltiplos utilizadores: o sistema deve suportar um mínimo de mil e quinhentos utilizadores ligados em simultâneo;
- Tempo de resposta: cada operação iniciada nos servidores deve ser processada em menos de dez segundos.

Manutenção

- Atualizações: o processo de atualização do sistema deve ser feito de forma a nunca interromper os seus serviços;
- Cópias de segurança: deve existir um plano de cópia e restauro da informação que garanta uma retenção de 30 dias das cópias efetuadas diariamente.

Restrições na arquitetura

- Integração com o portal SIG: as aplicações e módulos a desenvolver devem ser acoplados ao portal SIG existente;
- Navegadores de internet: as aplicações baseadas na *Web* terão de ser compatíveis com a maioria dos navegadores de Internet disponíveis.

5.3.2 Arquitetura da solução

Para dar resposta aos requisitos funcionais e não funcionais identificados nas secções anteriores, a equipa de projeto concebeu uma arquitetura baseada em componentes, que interagem entre si através de protocolos de comunicação bem definidos (ver Figura 15).

Assim sendo, os componentes idealizados para constituir a plataforma de notificação de problemas urbanos são:

- O portal SIG (no qual se incluem as aplicações setoriais e serviços associados);
- O servidor de bases de dados (com suporte ao processamento de dados espaciais);
- A aplicação móvel *Android*.

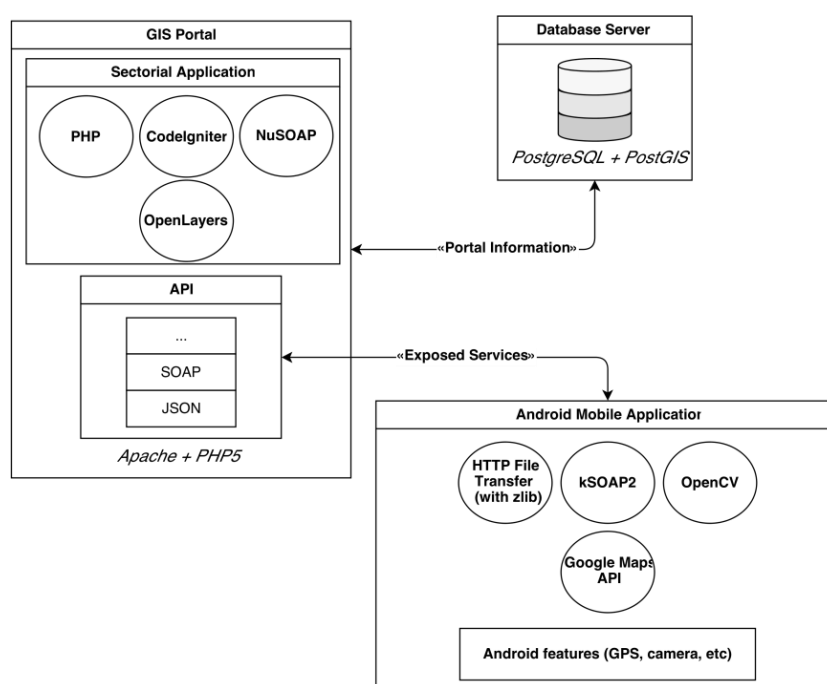


Figura 15 – Arquitetura da solução proposta [Santos et al., 2013]

O **portal SIG** engloba tanto o código base de suporte a toda a infraestrutura, como as aplicações dedicadas a um setor municipal específico e a camada de serviços para acesso por entidades externas. Desta forma, a reutilização da informação territorial e a herança de grande parte das funcionalidades básicas do portal estão asseguradas nas aplicações setoriais, assim como a garantia de interoperabilidade com outros sistemas de informação geográficos. Esta garantia de interoperabilidade só é possível devido à utilização de padrões de disponibilização de informação geográfica, como WMS e WFS, que permitem a aplicações externas consumir informação do sistema. O portal implementa estes padrões em ambos os sentidos, possibilitando não só a disponibilização de informação, como também o consumo de serviços remotos.

O **servidor de bases de dados** tem como principais funções a persistência da informação de toda a plataforma, o cálculo avançado de funções geométricas e a conversão da informação geométrica em formatos padrão. A utilização de uma extensão ao sistema de gestão de bases de dados para suporte a dados geométricos, possibilita a representação em tabela das coordenadas de objetos como pontos, linhas e polígonos, conferindo capacidades espaciais à base de dados. Desta forma, o armazenamento da informação em tipos de dados específicos, e a sua consulta recorrendo a funções otimizadas, minimizam os tempos de acesso tanto para leitura, como para escrita de informação espacial.

Devido ao número de acessos e frequência de operações de leitura e escrita de informação, este componente representa o *bottleneck*⁷ do sistema, necessitando de bastantes recursos computacionais.

A **aplicação móvel** para dispositivos *Android*, desenvolvida no âmbito deste projeto de investigação, é responsável pelo envio e consulta de relatórios de problemas urbanos pelos cidadãos de uma comunidade local. Para o efeito, este sistema interage com a plataforma através da camada de serviços disponibilizados pelo portal SIG, em conjunção com a componente de servidor da aplicação setorial de problemas urbanos (ver Figura 16). Estes serviços permitirão a consulta, inserção e manipulação da informação constante nas bases de dados do portal, estando acessíveis em métodos descritos no protocolo SOAP (Simple Object Access Protocol). A notação JSON será adotada para representar as transferências de estado e mensagens trocadas entre a aplicação móvel e a camada de serviços.

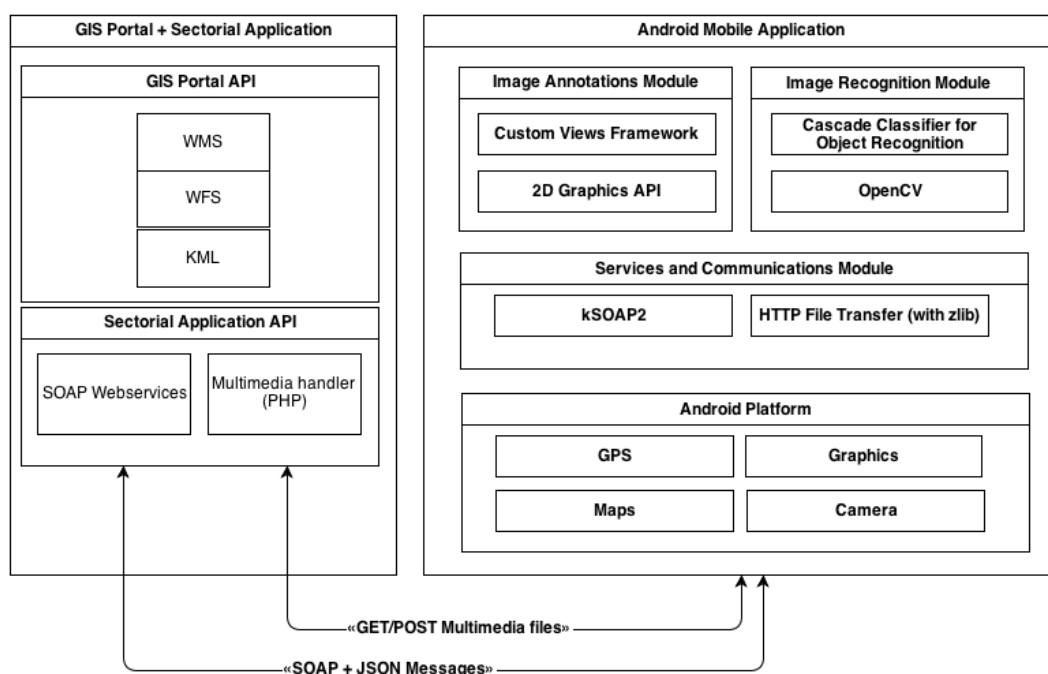


Figura 16 – Módulos constituintes da aplicação móvel

⁷ Componente no qual reside a limitação de performance e capacidade de um sistema

Sendo a componente móvel foco principal de toda esta investigação, de seguida serão apresentados e descritos em pormenor os módulos que a constituem: comunicação e serviços, anotação em imagens e reconhecimento de imagens.

5.3.2.1 Módulo de comunicação e serviços

O módulo de comunicação e serviços está encarregado da transferência de informação da aplicação *Android* para o servidor e vice-versa, formando uma camada de abstração simplificada para operações de envio e receção de multimédia como texto, imagens, áudio, e vídeo. Neste módulo foram utilizadas bibliotecas de programação com vista à otimização da largura de banda utilizada nas transferências, recorrendo à compressão dos dados no momento do envio [Isenhour, 2003].

Para a transferência de mensagens relativas aos objetos e estados associados, como informação de utilizadores e relatórios de notificações, é utilizado outro canal de comunicação: serviços *Web* no formato SOAP. Com recurso à biblioteca kSOAP2 para agilizar o processo de criação e interpretação de mensagens SOAP, as mensagens são passadas entre as aplicações cliente e servidor em formato textual seguindo a notação JSON.

5.3.2.2 Módulo de anotação em imagens

A anotação de imagens fica a cargo da API para gráficos 2D da plataforma *Android*, juntamente com a sua *framework* de elementos de interface gráfica. Estas tecnologias, agregadas num único módulo, possibilitam a criação de elementos gráficos próprios com suporte à demarcação manual de objetos em imagens. Esta demarcação, feita de forma natural pelo utilizador (desenhar com o dedo), é possível através da criação de linhas sobre uma imagem previamente carregada.

5.3.2.3 Módulo de reconhecimento de imagens

O módulo de reconhecimento de imagens recorre ao OpenCV, biblioteca de processamento de gráficos, para a implementação de um classificador de objetos em cascata. Este classificador atua sobre as imagens que se encontram associadas aos relatórios das ocorrências, devolvendo uma ou mais categorizações dependendo do tipo de objeto identificado. O módulo a desenvolver irá interagir com a biblioteca OpenCV, que deverá estar instalada nos dispositivos (a instalação prévia da biblioteca no dispositivo garante o máximo desempenho na execução do classificador).

Como a comunicação deste módulo com o OpenCV é feita através da invocação de um serviço, o utilizador pode atualizar a biblioteca OpenCV (que poderá até ser partilhada por outras aplicações) para uma versão mais recente.

5.4 Implementação

A secção que se segue pretende documentar os processos relacionados com o desenvolvimento da solução proposta, desde a fase de identificação e integração de tecnologias, à escrita de algoritmos específicos para tratamento de imagem.

Em primeiro lugar são apresentadas as tecnologias utilizadas no projeto, em conjunto com a justificação da sua escolha de integração na solução. De seguida, é introduzido o modelo de persistência de informação, começando por capturar as interações entre as entidades intervenientes no sistema através de um diagrama entidade-relacionamento, passando depois para a explanação do modelo de dados concebido. Ainda na categoria de modelação de persistência, é dado ênfase ao componente para tratamento de informação espacial, apresentando os conceitos-chave de georreferenciação implementados.

Segue-se a definição da camada de serviços, componente responsável pela tradução dos pedidos efetuados pelas aplicações externas em invocações à API da plataforma. Esta camada introduz um nível de indireção no acesso à base de dados, com recurso à exposição de métodos para a *Web* seguindo a especificação WSDL (para disponibilização dos serviços) e SOAP (formato usado na troca de mensagens entre sistemas). Nesta subsecção são ainda apresentados os métodos que fazem parte do protótipo no momento da escrita deste documento.

Posteriormente à descrição da camada de serviços é apresentada a aplicação móvel da plataforma, detalhando questões gerais de implementação como a integração das bibliotecas necessárias, a organização e estruturação do código (inclusão de padrões de engenharia de *software*) e o aspeto da interface gráfica com o utilizador. São ainda revistos em pormenor três módulos constituintes da aplicação móvel, que se evidenciam pelo seu grau de inovação em plataformas móveis: o módulo de comunicação e serviços, anotações em imagem e reconhecimento de objetos em imagens. Para cada um destes módulos, é feita uma análise do problema a que dão resposta, enquadrando-o devidamente nas temáticas abordadas nesta investigação, e recorrendo a diagramas de classes e explicação dos algoritmos e fórmulas essenciais para enriquecer a demonstração das capacidades.

No decorrer da apresentação dos vários módulos, são utilizadas imagens da aplicação para ilustrar os desenvolvimentos efetuados, fornecendo uma perspetiva prática sobre o seu funcionamento.

5.4.1 Tecnologias utilizadas

O desenvolvimento de uma solução de notificação de problemas urbanos envolve a correta definição e seleção das tecnologias a utilizar, sob pena de comprometer o sucesso do projeto. Neste sentido, as ferramentas apresentadas de seguida são o resultado de uma análise profunda da oferta tecnológica disponível no mercado, constituindo as soluções mais proeminentes e inovadoras nos seus ramos de negócio.

Para o desenvolvimento do código base do portal foi selecionada a **linguagem PHP**, por forma a usufruir da quantidade de *frameworks*, módulos, e extensões disponíveis, além da crescente comunidade de programadores que contribui para a evolução da linguagem. Sendo uma tecnologia escalável, eficiente, dinâmica e multiplataforma, a implementação do portal SIG foi efetuada a um ritmo elevado, permitindo instalações e configurações independentes do sistema operativo.

Escolhida a tecnologia para suportar a execução do lado do servidor, foi necessário optar por uma que apresentasse graficamente a informação espacial da base de dados. O **OpenLayers** foi a solução adotada para o lado do cliente, permitindo o carregamento de pontos, linhas e polígonos num mapa, através da utilização de padrões como KML e GeoJSON. Esta biblioteca de código aberto possibilita a apresentação de mapas de diferentes fornecedores como OpenStreetMap, Bing e Google, adicionando uma camada de tradução das funções das várias APIs, tornando transparente a interação com o mapa.

O SGBD (sistema gestor de bases de dados) utilizado neste projeto dá pelo nome de **PostgreSQL**. Atualmente na versão nove, esta solução apresenta-se como o sistema de bases de dados de código aberto mais avançado do mundo, garantindo fiabilidade e segurança totais no armazenamento da informação. Este sistema de bases de dados objeto-relacionais permite, através do seu sistema de extensões, integrar o módulo **PostGIS** para adição de capacidades espaciais. Este módulo agrega inúmeras funções para processamento geométrico, como interseção de pontos, cálculo de distâncias e conversão entre sistemas de coordenadas, constituindo a solução de código aberto mais completa para bases de dados espaciais.

O **nuSOAP** foi a escolha da equipa do projeto para o desenvolvimento dos serviços *Web* no protocolo SOAP. Como esta solução foi construída na linguagem PHP, a integração no portal SIG ficou facilitada, permitindo a invocação direta de métodos da camada de acesso à base de dados.

Na componente móvel, foi utilizada a **plataforma Android** juntamente com os seus módulos de manipulação de gráficos, protocolos de comunicação, sensores e mapas. A escolha deste sistema operativo para acolher a primeira versão da aplicação móvel prende-se com o volume crescente de dispositivos e utilizadores neste segmento de mercado, tal como evidenciado no capítulo 2.

A necessidade da inclusão da tecnologia **OrmLite** surge com a recorrente falha de conectividade dos dispositivos móveis, obrigando à manutenção de cópias locais da

informação mais relevante. Aliada a esta situação está também a dificuldade de sincronização da informação local dos dispositivos com o portal SIG, resultando em problemas de consistência e integridade dos dados. O OrmLite permite, através de uma API minimalista, efetuar o mapeamento de classes de objetos para tabelas de uma base de dados SQLite⁸, persistindo a informação no dispositivo móvel.

O **kSOAP2** é o nome da solução para a criação, envio, receção e interpretação de mensagens descritas no protocolo SOAP, integrada neste projeto. Através da sua API, é possível ao programador manipular as mensagens desconhecendo os conceitos e detalhes da especificação formal do protocolo, agilizando a tarefa de consumir serviços disponíveis na *Web*.

No módulo de reconhecimento de imagem foi utilizada a plataforma **OpenCV**, líder no ramo da computação gráfica e aprendizagem automática de sistemas. Esta tecnologia suporta o desenvolvimento de classificadores de objetos, possibilitando o reconhecimento dos problemas urbanos mais comuns. Com vista a maximizar o desempenho dos algoritmos desenvolvidos, foi utilizado o **NVIDIA Tegra Toolkit**, uma vez que inclui otimizações e melhoramentos para dispositivos equipados com processadores NVIDIA Tegra [Nvidia, 2013].

5.4.2 Base de Dados

A modelação de persistência dos dados careceu de especial atenção na conceção desta solução, uma vez que o servidor de bases de dados é um dos pontos nevrálgicos do sistema, concentrando em si a maior parte dos acessos para leitura e escrita. Para maximizar o desempenho e integridade da informação, foi construído um modelo de dados normalizado que permite mapear entidades de domínio em tabelas e registos de bases de dados. A construção do modelo de dados sustentou-se no resultado das tarefas de análise e desenho do sistema (ver secção 5.3 - Modelo proposto), identificando as entidades e atributos indispensáveis para persistência e futura consulta de informação.

5.4.2.1 Diagrama Entidade-Relacionamento

A definição da informação a persistir foi conseguida através da construção de um diagrama Entidade-Relacionamento (ver Figura 17), permitindo a representação visual das entidades e das suas relações:

⁸ SQLite – sistema que suporta bases de dados transacionais sem necessidade de configuração ou servidores dedicados [SQLite, 2013]

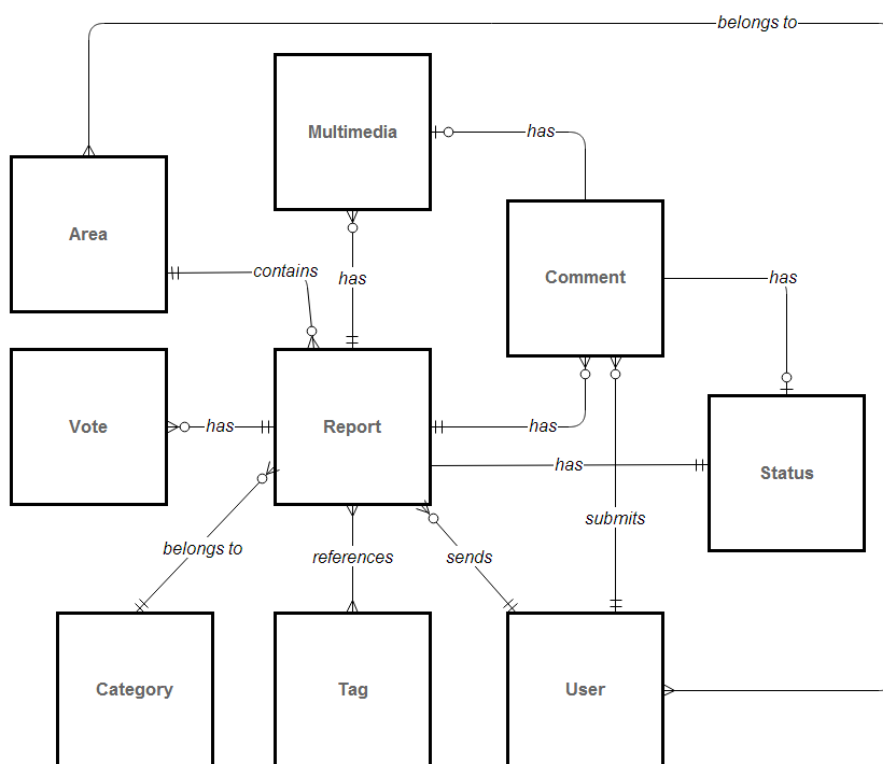


Figura 17 – Diagrama de Entidade-Relacionamento (Notação *Crow's Foot*⁹)

Neste diagrama foram identificadas as seguintes entidades:

- Relatório: entidade que representa a notificação de uma ocorrência submetida por um utilizador. Pode ter comentários, votos, e tags associados, assim como um estado de resolução, uma área de abrangência e uma categorização;
- Utilizador: abrange todos os indivíduos que utilizam a plataforma para a notificação de problemas. Pode submeter relatórios e enviar comentários, e fazer parte de uma ou mais comunidades locais;
- Área: define uma zona de interesse para a captura de relatórios de notificações. Refere-se tipicamente a uma delimitação geográfica de um município ou comunidade local;
- Multimédia: representa um elemento multimédia no sistema. É utilizado para associar anexos como imagens, áudio e vídeo a relatórios e comentários;
- Comentário: elemento associado a relatórios que fomenta a discussão dos problemas notificados entre os cidadãos da comunidade;

⁹ Crow's Foot – Notação para diagramas Entidade-Relacionamento que adota um elemento gráfico semelhante a uma pata de corvo para representação de relações "muitos para muitos".

- Estado: entidade encarregada da atribuição de estados de resolução a relatórios e comentários;
- Voto: representa a prioridade e ordem de resolução das ocorrências indicadas pelos utilizadores;
- Categoria: define classes de problemas urbanos. Permite enquadrar uma ocorrência e facilitar os processos de triagem, encaminhamento e resolução;
- *Tag*: entidade que permite associar palavras-chave a relatórios de ocorrências.

Este diagrama sustentou a construção do modelo de dados do sistema, permitindo identificar as relações existentes entre as entidades e efetuar o mapeamento para tabelas e atributos (ver Anexo I).

5.4.2.2 Componente espacial

A inclusão da extensão PostGIS conferiu ao sistema gestor de bases de dados as potencialidades de manuseamento de informação espacial necessárias para armazenar dados geográficos em tabelas de forma eficiente, fiável e segura. Além da adição de novos tipos de dados e atributos às bases de dados, esta extensão contém um vasto conjunto de funções auxiliares que permitem efetuar operações e cálculos complexos sobre a informação georreferenciada. Estas funções, tendo o desempenho como principal objetivo, são executadas diretamente sobre os registos do SGBD, eliminando a necessidade da sua transferência para bibliotecas de tratamento de informação espacial.

Nesta perspetiva, a equipa de projeto decidiu concentrar todo o processamento intensivo de informação geométrica no componente da base de dados, desenvolvendo funções otimizadas para tarefas como o cálculo de distâncias, intersecções de pontos e polígonos e conversão entre sistemas de coordenadas. A função para determinação das ocorrências a devolver aos utilizadores (ver Figura 18) é uma das funções implementadas na íntegra neste componente:

```

1: //Algoritmo de seleção das notificações da área de interesse do utilizador
2: Entrada: latitudeUtilizador, longitudeUtilizador, raio, idUtilizador (opcional)
3: Saída: ListaOcorrências
4: Início
5: ListaOcorrências <- [];
6: //Verificar se foi enviado um identificador de utilizador válido
7: Se VerificaIDUtilizador(idUtilizador) Então
8: //Obtém a lista de áreas que o utilizador subscreveu ou pertence
9: ÁreasUtilizador <- ObterListaÁreasUtilizador(idUtilizador);
10: //Percorre cada uma das áreas e obtém a lista de ocorrências dessa área
11: Para cada ÁreaUtilizadori Em ÁreasUtilizador Faz
12: //Soma as ocorrências obtidas à lista de ocorrências atual
13: ListaOcorrências <- ListaOcorrências + ObterOcorrênciasÁrea(ÁreaUtilizadori);
14: FimPara
15: FimSe
16://Percorre todas as ocorrências e adiciona as ocorrências mais próximas do utilizador
18: Para cada Ocorrênciai Em TotalOcorrências Faz
19: Se CalcularDistancia(Ocorrênciai.lat, Ocorrênciai.lon, latitudeUtilizador, longitudeUtilizador) < raio Faz
20: ListaOcorrências <- Reúne(ListaOcorrências, Ocorrênciai);
21: FimSe
22: FimPara
23: Devolver ListaOcorrências
24: Fim

```

Figura 18 – Algoritmo de determinação dos relatórios de ocorrências a apresentar

Este algoritmo evidencia o mecanismo adotado para a limitação das ocorrências a devolver aos utilizadores. Numa primeira fase, verifica se o utilizador está registado na plataforma; em caso afirmativo, é obtida a lista de áreas que subscreveu para receção de notificações (por exemplo, a sua área de residência e a área circundante do seu local de trabalho). Para cada uma das áreas, são determinadas as ocorrências inseridas nas suas delimitações geográficas, e adicionadas a uma lista comum. De seguida, a lista de todas as ocorrências da plataforma é percorrida, a fim de determinar as ocorrências inseridas num raio de pesquisa indicado por parâmetro. Essas ocorrências são também adicionadas à mesma lista comum, que é por fim devolvida ao utilizador.

As operações para cálculo da distância e determinação das ocorrências de uma área utilizam procedimentos e funções otimizadas da biblioteca PostGIS, acelerando o processamento da informação e consequente devolução dos resultados.

5.4.3 Camada de serviços

A camada de serviços introduz um nível de indireção no acesso à base de dados do sistema, permitindo a realização de operações de leitura e escrita sobre a informação do domínio da aplicação setorial. Esta camada complementa a oferta de serviços disponibilizada de raiz pelo portal SIG, como a obtenção de pontos de interesse através de WFS e KML, adicionando métodos para consulta, inserção, e atualização de relatórios de ocorrências.

Para a publicação e disponibilização destes métodos, foi utilizado o padrão WSDL (Web Services Definition Language), que permite descrever as operações suportadas indicando em cada caso o nome, parâmetros necessários para a invocação e o tipo de dados de retorno. Tanto o envio como a receção das mensagens trocadas entre a API e os seus clientes estão assegurados pelo protocolo de transporte HTTP (Hypertext Transfer Protocol), sendo que os pedidos de invocação e consequentes respostas respeitam o formato SOAP.

5.4.3.1 Integração da tecnologia nuSOAP

A biblioteca nuSOAP é uma solução para a criação e publicação de serviços *Web* baseados nas especificações WSDL e SOAP. Escrita na linguagem PHP, esta biblioteca agrega funções para a correta definição do WSDL e formatos de mensagens SOAP necessários, encapsulando a complexidade destes protocolos em processos simples.

Devido à necessidade de integração transparente de uma tecnologia de serviços *Web* com o portal SIG, a equipa do projeto acordou a inclusão desta biblioteca na plataforma, aproveitando todas as facilidades inerentes à manutenção da mesma tecnologia e linguagem de programação.

5.4.3.2 Operações implementadas

O ritmo de adição das funções nesta camada acompanhou os desenvolvimentos efetuados na componente móvel, permitindo construir desde cedo versões demonstráveis da solução.

No momento da escrita deste documento, o protótipo desenvolvido conta já com as seguintes operações:

- **Adicionar notificação:** responsável pela adição de uma nova notificação no sistema. Requer a passagem de parâmetros como o título, descrição, *tags*, categoria, localização geográfica e identificador do utilizador. Retorna o identificador do relatório inserido, ou um código de erro caso a operação não se concretize;
- **Obter notificações:** devolve a lista de notificações de acordo com as áreas a que o utilizador pertence, somando às notificações próximas da sua posição geográfica;
- **Obter detalhes de uma notificação:** retorna a informação correspondente a uma notificação específica;
- **Obter comentários de uma notificação:** obtém a lista de comentários associados a uma notificação, juntamente com a informação dos utilizadores que participaram na discussão;
- **Obter votos de uma notificação:** obtém a lista de votos associados a uma notificação, juntamente com a informação dos utilizadores que efetuaram esses mesmos votos.

5.4.4 Aplicação móvel

O foco desta investigação incidiu sobre a componente móvel, uma vez que esta constitui o maior desafio nas várias temáticas abordadas, e sobre a qual existem poucos desenvolvimentos efetuados até ao momento.

A aplicação móvel, desenvolvida na plataforma *Android*, utiliza as bibliotecas nativas do sistema operativo para atingir o máximo desempenho, e conta com a integração de tecnologias robustas e inovadoras para a realização de tarefas que o S.O., por si só, tem dificuldades em executar.

Numa perspetiva de abranger a maioria dos dispositivos móveis com este sistema operativo, a aplicação foi desenvolvida com vista a manter a compatibilidade com a plataforma *Android Froyo* (versão da API 2.2) e seguintes.

5.4.4.1 Integração de bibliotecas

O primeiro passo na construção da aplicação móvel passou pela integração das bibliotecas necessárias, validando assim a abordagem tomada e garantindo a possibilidade de implementação de todos os módulos.

Para colmatar os problemas e dificuldades encontradas na secção 5.3, foram integradas as seguintes bibliotecas:

- **kSOAP2**: esta biblioteca encapsula o protocolo SOAP em funções simples, possibilitando a troca de mensagens e chamada de procedimentos de máquinas remotas;
- **Asynchronous Http Client**: utilizada em aplicações como Instagram ou Pinterest, esta biblioteca transforma os pedidos HTTP em chamadas assíncronas, conferindo características responsivas às aplicações;
- **ORMLite**: efetua o mapeamento de objetos da aplicação para tabelas de uma base de dados SQLite, fornecendo métodos para a criação, leitura, atualização e eliminação de registos;
- **Universal Image Loader**: biblioteca utilizada para a transferência e visualização das imagens dos relatórios de notificações;
- **OpenCV**: plataforma utilizada para efetuar a deteção e classificação dos problemas urbanos mais comuns.

A integração de bibliotecas na plataforma *Android* revela-se um processo simples, bastando incluir os executáveis numa pasta dedicada à compilação de bibliotecas externas. A aplicabilidade deste procedimento verificou-se nas bibliotecas kSOAP2, Asynchronous Http

Client, ORMLite e Universal Image Loader, não sendo executável na plataforma OpenCV. Neste caso, devido a esta biblioteca ser desenvolvida com o intuito de atingir o máximo desempenho possível, inclui código otimizado para as várias arquiteturas de processadores *Android*, o que obriga à inclusão de executáveis dependentes do dispositivo. Para solucionar este problema, a equipa responsável pelo OpenCV desenvolveu o *OpenCV Manager*, aplicação móvel para efetuar a descarga, instalação e gestão automática das versões do OpenCV compatíveis com o dispositivo. Esta abordagem permite que várias aplicações partilhem o mesmo conjunto de bibliotecas OpenCV, eliminando por um lado o desperdício de espaço associado à duplicação de executáveis, e por outro a necessidade da manutenção de várias versões da mesma tecnologia. Esta filosofia requer que as aplicações se liguem a um serviço do *OpenCV Manager* para invocar métodos da biblioteca, sendo que a devolução dos resultados é feita de forma assíncrona.

5.4.4.2 Organização do código

A estruturação cuidada do código conduz à compreensão clara do propósito de cada um dos componentes, além de contribuir para uma melhor sustentabilidade e longevidade do projeto. Nesta aplicação foram adotadas boas práticas no desenvolvimento de soluções para dispositivos móveis, como a adaptação a vários tamanhos de ecrã e poder de processamento dos dispositivos, assim como implementados padrões de programação orientada a objetos para resolução de problemas típicos da engenharia de aplicações.

O seguimento rigoroso destas diretivas levou à divisão lógica das classes da aplicação pelos seguintes pacotes:

- **Acesso a dados:** agrega classes responsáveis pela implementação das interfaces de acesso à informação da aplicação, por intermédio de serviços *Web*. Transforma a informação proveniente dos serviços em objetos do modelo compreendidos pela aplicação móvel;
- **Base de dados:** conjunto de classes encarregadas de efetuar a persistência da informação proveniente dos serviços numa base de dados SQLite;
- **Eventos:** contém a definição dos eventos despoletados pela aplicação, como chegada de informação relativa a relatórios e multimédia, ou obtenção de um sinal GPS válido;
- **Exceções:** conjunto de exceções lançadas pela aplicação para diferenciação das situações de erro, como por exemplo se o servidor de relatórios está inalcançável;
- **Interface gráfica:** agrega as classes responsáveis pela construção da interface gráfica da aplicação, e da interação dos componentes com o utilizador;
- **Modelo:** contém as classes que definem os objetos de domínio compreendidos pela aplicação, como relatórios de notificações, comentários, *tags*, utilizadores, etc.;

- **Utilidades:** agrega as classes que não pertencem a nenhuma divisão lógica específica, mas são necessárias para efetuar tarefas secundárias, como conversão de texto de e para o formato base64;
- **Interação com sensores:** responsável pela interação de baixo nível com os sensores do dispositivo, como GPS, sensor de proximidade, bússola, etc.;
- **Rede:** conjunto de classes que garante a comunicação a baixo nível do tráfego HTTP para submissão de multimédia, e a troca de mensagens SOAP com o servidor;
- **Serviços:** implementação de uma camada de serviços do sistema operativo para suportar operações assíncronas nas tarefas de execução demorada ou dependentes de terceiros, como a determinação do nome de uma rua a partir da localização GPS, ou a obtenção e submissão de relatórios de notificações;
- **OpenCV:** agrega as classes encarregadas da comunicação com o serviço do OpenCV *Manager*, assim como a implementação do modelo de reconhecimento de problemas urbanos.

5.4.4.3 Aspeto gráfico

A interface gráfica da aplicação seguiu rigorosamente as indicações de desenho e usabilidade para a plataforma *Android*, permitindo adaptar-se a vários tipos e tamanhos de ecrã, assim como a dispositivos de diversas gamas de desempenho. A adoção destas regras torna as aplicações responsivas às ações dos utilizadores, minimizando o seu esforço na realização de tarefas e conduzindo à sua satisfação e vontade de utilizar o sistema.

A disponibilização da aplicação em vários idiomas também é um ponto de especial relevância, uma vez que é um fator de abandono das aplicações por parte dos utilizadores. A rapidez de resposta devido à implementação em código nativo do sistema operativo, aliada à implementação de elementos que sigam paradigmas de interação familiares ao utilizador, como botões, separadores e listas ativados pelo toque, levam à imediata compreensão de como navegar no sistema. A Figura 19 ilustra as especificações de *design* e usabilidade implementadas ao longo de toda a aplicação:

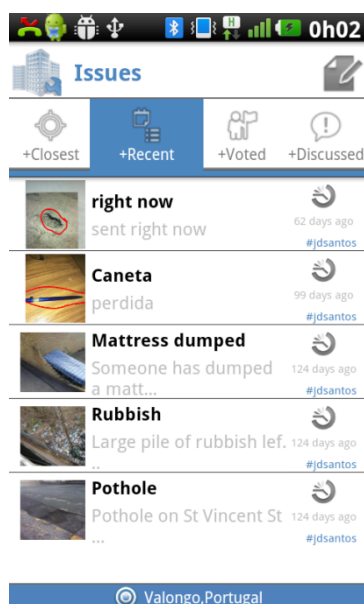


Figura 19 – Aplicação móvel – Listagem de ocorrências

Neste ecrã é possível verificar a existência de uma navegação por separadores, em que cada separador apresenta uma lista de notificações ordenada segundo parâmetros pré-estabelecidos. A barra de título permite navegar para a janela principal da aplicação, assim como iniciar a escrita de um novo relatório de uma ocorrência. O acesso ao detalhe de uma notificação é feita por intermédio de toque simples na linha da notificação que se pretende visualizar. Estes controlos facilitam a consulta rápida de notificações, permitindo realizar ações individuais através do ecrã de detalhe, ou instintivamente incitar o preenchimento de um novo relatório de uma ocorrência. Foi também implementado um menu de contexto dependente do ecrã atual, tornando o acesso a opções específicas ainda mais expedito.

As secções seguintes pretendem descrever em detalhe o processo de desenvolvimento de cada um dos principais módulos constituintes da aplicação, aprofundando a documentação sobre a aplicação móvel efetuada nesta secção.

5.4.5 Módulo de comunicação e serviços

O módulo de comunicação e serviços está encarregado de efetuar o carregamento da informação na aplicação, através da transferência dos dados dos servidores da plataforma de notificação de problemas urbanos. Este módulo desempenha duas funções em simultâneo: por um lado, gere as conexões com os servidores da solução; por outro, disponibiliza serviços que realizam as transferências de forma assíncrona, para que a aplicação se mantenha responsiva.

Este foi o primeiro módulo a ser desenvolvido, uma vez que a consulta e submissão de relatórios de notificações são funcionalidades básicas deste tipo de sistemas. A utilização das bibliotecas identificadas na secção “Tecnologias utilizadas”, permitiu a implementação deste módulo recorrendo a um conjunto de classes minimalista (ver Figura 20):

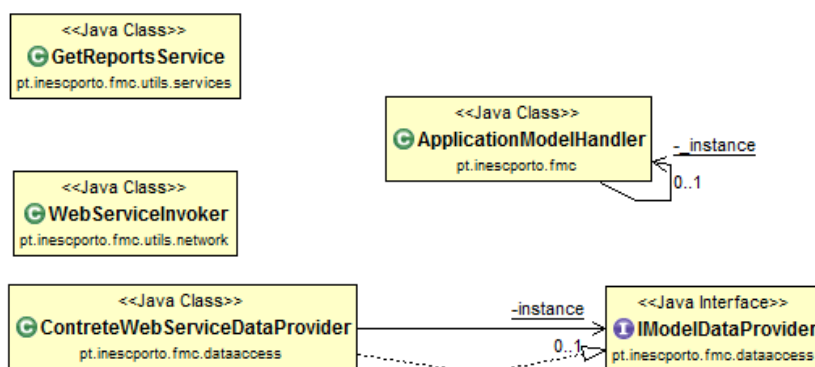


Figura 20 – Módulo de serviços e comunicação – Diagrama de classes

Neste diagrama exemplificativo (sem a representação de todos os serviços, nem das operações e atributos das classes) é possível verificar os relacionamentos diretos de herança e composição entre as várias classes de objetos, não estando patente a interação existente por instanciação.

A classe “GetReportsService” implementa um serviço que torna a operação de descarga dos relatórios de ocorrências totalmente assíncrona. Para isso, esta classe recorre ao “ApplicationModelHandler”, um objeto que efetua a transição dos objetos devolvidos pelo “ConcreteWebServiceDataProvider” para a lógica da aplicação. O “WebServiceInvoker” lida com as mensagens SOAP retornadas pelos servidores a um baixo nível, e processa a informação para passar ao “ConcreteWebServiceDataProvider”. Toda a lógica de negócio associada aos objetos devolvidos está concentrada no “ApplicationModelHandler”, ponto nevrálgico do sistema em que é efetuado o mapeamento dos objetos e a preservação do estado global da aplicação. Esta classe está também encarregada de delegar, nas classes dedicadas à persistência dos dados, a tarefa de armazenar a versão atualizada dos objetos na base de dados.

5.4.6 Módulo de anotações em imagem

O processo de desenvolvimento de uma tecnologia de anotação de objetos em imagens envolveu a definição e construção de um novo componente de interface gráfica para o sistema operativo *Android*. Este componente, idealizado através da observação de outros elementos que compõem a interface gráfica do sistema operativo, permite que os cidadãos realcem a zona problemática de uma imagem reportada através de um método de interação natural: o toque.

O processo de desenvolvimento deste módulo sustentou-se na análise efetuada no capítulo 3 que sugere a utilização das bibliotecas da plataforma *Android* para manipulação de gráficos 2D. O componente implementado assenta sobre a classe base “Canvas”, responsável pelo desenho de gráficos 2D em superfícies, em conjunção com um objeto “ImageView” para apresentação das imagens de notificação sobre as quais se deseja anotar (ver Figura 21):

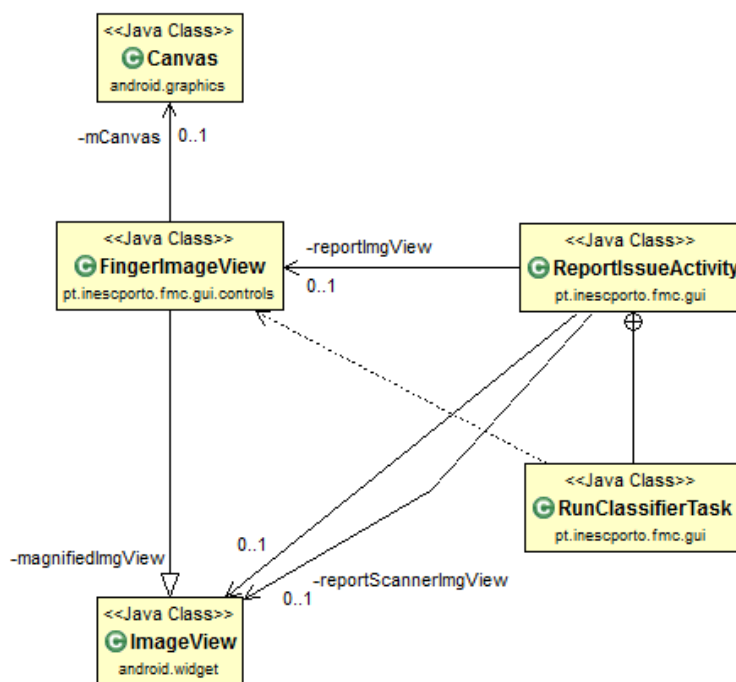


Figura 21 – Módulo de anotações em imagem – Diagrama de classes

O controlo “FingerImageView”, representado como uma classe de objetos, herda todas as capacidades de um controlo “ImageView”, permitindo-lhe apresentar as imagens anexas aos relatórios de notificações. Para atingir a demarcação de objetos numa imagem de forma natural, foi adicionado o objeto “Canvas” da API de gráficos 2D da plataforma *Android*, que possibilita a criação de uma área de desenho livre no ecrã. Neste módulo, a área do objeto “Canvas” é gerada dinamicamente consoante o tamanho da imagem a apresentar e colocada sobre esta, sobrepondo-a. A partir deste momento, todos os eventos associados ao toque na

imagem são capturados pelo objeto “Canvas”, que se encarrega de desenhar a demarcação efetuada pelo utilizador (ver algoritmo da Figura 22):

```

1: //Algoritmo de desenho de linhas numa imagem
2: Entrada: accaoUtilizador,posXLinhaInicial,posYLinhaInicial,posXLinhaFinal,posYLinhaFinal,
   caminhoLinhas, toleranciaMov
3: Saída:
4: Inicio
5: //Verificar o tipo de acção do utilizador
6: Se accaoUtilizador=PRESSIONAR Então
7:   //Se foi iniciado um desenho, mover a caneta para a posição do ecrã tocada
8:   caminhoLinhas <- [];
9:   caminhoLinhas <- MoverPara(caminhoLinhas,posXLinhaFinal,posYLinhaFinal);
10:  posXLinhaInicial <- posXLinhaFinal;
11:  posYLinhaInicial <- posYLinhaFinal;
12: Senão Se accaoUtilizador=MOVER Então
13:   //Se o utilizador moveu o dedo ao longo do ecrã, guardar as posições para desenhar
14:   dX <- ABS(posXLinhaFinal-posXLinhaInicial);
15:   dY <- ABS(posYLinhaFinal-posYLinhaInicial);
16:   //Se a diferença da posição inicial para a final é maior do que a tolerância definida, calcular nova posição
17:   Se dX >= toleranciaMov OU dY >= toleranciaMov Então
18:     //Calcular uma curva de Bezier para suavizar o desenho das linhas
19:     caminhoLinhas <- caminhoLinhas + CalcularCurvaQuadraticaBezier(posXLinhaInicial,posYLinhaInicial,
   (posXLinhaFinal + posXLinhaInicial) / 2,(posYLinhaFinal + posYLinhaInicial) / 2);
20:     posXLinhaInicial <- posXLinhaFinal;
21:     posYLinhaInicial <- posYLinhaFinal;
22:   FimSe
23: Senão Se accaoUtilizador=LARGAR Então
24:   //Se o utilizador largou o dedo do ecrã, é efetuado o desenho do caminho de linhas calculado
25:   caminhoLinhas <- caminhoLinhas + Ponto(posXLinhaInicial,posYLinhaInicial);
26:   DesenharCaminho(caminhoLinhas);
27:   caminhoLinhas <- [];
28: FimSe
29: Fim
30: Fim

```

Figura 22 – Módulo de anotações em imagem – Algoritmo para desenho de linhas

Este algoritmo ilustra o ciclo de eventos associado ao componente de demarcação desenvolvido. Nele, é possível identificar os três tipos de interação do utilizador com o componente: pressionar (iniciar o desenho), mover (continuar o desenho) e largar (terminar o desenho).

Na fase inicial da interação, que corresponde ao utilizador pressionar o ecrã do dispositivo, são guardadas as coordenadas X e Y da posição pressionada, e o apontador de início do desenho é movido para essa localização.

Aquando da realização de um movimento, a decisão das linhas a considerar é tomada tendo em conta um valor de controlo de tolerância (permite que o utilizador desenhe linhas mais ou menos retas sem grande precisão no gesto). Para isso, em primeiro lugar é calculada a distância efetiva entre os pontos recorrendo à fórmula do valor absoluto:

$$|x| \begin{cases} x, & x \geq 0 \\ -x, & x \leq 0 \end{cases} \quad (1)$$

, onde x representa um valor no eixo X ou Y das coordenadas de um ponto.

De seguida, é determinado se a distância obtida excede ou não o valor de controlo. Em caso afirmativo, essa linha é considerada no desenho, e é calculada uma curva quadrática de Bézier para suavizar o desenho das linhas:

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, t \in [0,1] \quad (2)$$

,onde $B(t)$ é a função que descreve a curva de Bézier ao longo do tempo, t um instante de tempo e P_0 e P_1 os pontos das extremidades da reta.

O fim da interação com o componente, que corresponde ao momento em que o utilizador retira o dedo do ecrã, espoleta o procedimento de desenho das linhas sobre a imagem, resultando no efeito da Figura 23:



Figura 23 – Aplicação móvel – Demarcação manual de patologia na estrada

Findo o desenho das linhas na imagem, o utilizador tem à sua disposição controlos para guardar ou limpar a demarcação efetuada, consoante a anotação corresponda ou não às suas expectativas. Caso as alterações sejam guardadas, é gerada uma nova imagem com base na junção da fotografia original do problema e da anotação efetuada, sendo automaticamente anexa ao relatório da ocorrência.

5.4.7 Módulo de reconhecimento de imagem

A natureza inconstante da detecção e classificação de objetos em imagens, tornou a implementação do módulo de reconhecimento um desafio digno de referência, exigindo a investigação, análise e adaptação de algoritmos para corresponder à realidade das plataformas móveis. De entre as inúmeras dificuldades encontradas, a versatilidade necessária no algoritmo para consciencialização e adaptação ao ambiente que rodeia os objetos a detetar, e o requisito de máxima fluidez na sua execução, foram as que mais se demarcaram.

A escolha da plataforma OpenCV, suportada pelo estudo pormenorizado das tecnologias de detecção e classificação do capítulo 4, permitiu a definição de um algoritmo de carregamento de classificadores em cascata para categorização das fotografias dos problemas urbanos mais comuns. A organização destes classificadores segundo um modelo em cascata procura subdividir a complexidade em vários níveis ou estádios, iniciando o processo de detecção pelo classificador mais simples. Esta disposição tende a minimizar o tempo de detecção dos objetos, uma vez que áreas de uma imagem que não revelem particular interesse para o classificador mais básico, não podem nunca constituir possíveis candidatos para reconhecimento pelos classificadores mais elaborados [Viola & Jones, 2001].

Os classificadores são alimentados com regiões de interesse determinadas segundo o método *Haar*, usufruindo de todas as vantagens de uma análise com conceitos do domínio afetos às zonas, ao invés da análise direta sobre os píxeis das imagens [Viola & Jones, 2001]. Este método, apesar de representar um custo computacional mais elevado, foi selecionado em detrimento da alternativa LBP (*Local Binary Patterns*), devido a esta não ser tão fiável na identificação das regiões a processar. A técnica *Haar* efetua um varrimento das imagens seguindo os princípios de uma janela deslizante, submetendo cada fragmento ao processamento dos classificadores por ordem crescente de complexidade. No entanto, este varrimento efetua uma triagem deficiente das regiões de interesse a considerar, devolvendo um elevado conjunto de zonas que torna impraticável o seu processamento em tempo útil. Para solucionar este problema, a decisão de quais as regiões de interesse a utilizar na aprendizagem e treino do algoritmo, fica a cargo do *Gentle AdaBoost* (Adaptive Boost), o método de seleção e aprendizagem de zonas de interesse mais eficiente segundo o estudo empírico efetuado em [Lienhart et al.,2003].

Devido à escassez de documentação e exemplos concretos para dispositivos móveis, o desenvolvimento deste módulo sustentou-se na reescrita de algoritmos da biblioteca OpenCV originalmente destinados a plataformas *desktop*.

O modelo de reconhecimento concebido (ver Figura 24) assenta sobre as potencialidades da classe “*CascadeClassifier*” da biblioteca OpenCV, permitindo a execução dinâmica de classificadores de objetos:

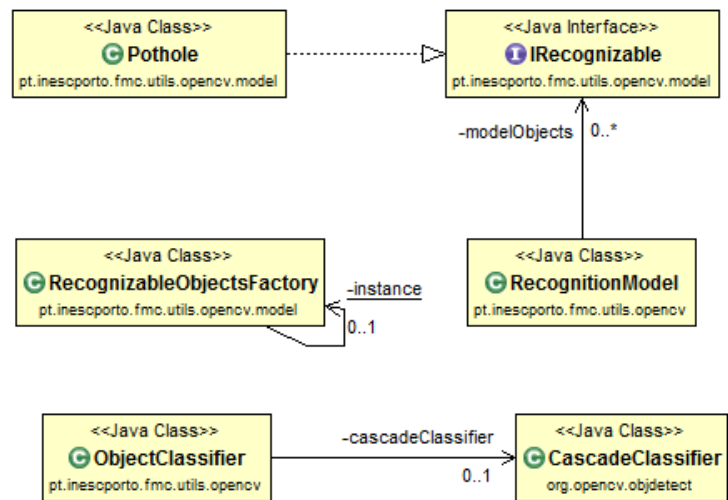


Figura 24 – Módulo de reconhecimento de imagem – Diagrama de classes

A interface “IRecognizable” define comportamentos que objetos passíveis de serem reconhecidos na plataforma necessitam de implementar, como a devolução do ficheiro XML (eXtensible Markup Language) que representa o classificador, e o conjunto de *tags* que caracterizam o objeto identificado. As instâncias desta classe são criadas através da “RecognizableObjectsFactory”, e mantidas no modelo de reconhecimento (denominado de “Recognition Model”), para posteriormente serem utilizadas na deteção e classificação das imagens. Esta estruturação permite que o reconhecimento de novos objetos pelo sistema possa ser feito com relativa facilidade, bastando construir uma nova classe que implemente a interface “IRecognizable” e adicionar uma instância ao modelo de reconhecimento.

O protótipo desenvolvido no âmbito deste projeto implementa a deteção automática de um tipo de problema urbano, procurando facilitar a tarefa dos utilizadores na classificação das ocorrências a reportar. Constitui também um objetivo deste protótipo validar a abordagem tomada na construção desta solução, e assim servir de modelo para a inclusão de novas classificações de objetos. O tipo de problema urbano selecionado para implementação e estudo foi “Buracos na estrada”, por observação da ocorrência frequente deste tipo nas plataformas atuais de notificação de problemas cívicos pela equipa do projeto, e por constituir um maior desafio na tarefa de deteção e classificação, devido à diversidade de situações e combinações possíveis [Koch & Brilakis, 2011]. A classe “Pothole” representa esta categoria de problemas urbanos no sistema, constituindo um caso de estudo para avaliação da fiabilidade e aplicabilidade dos algoritmos de classificação em dispositivos móveis, capturado na secção 5.5.

A classe “ObjectClassifier” agrega toda a lógica inerente à interação com a plataforma OpenCV, desde o carregamento de ficheiros de descrição de classificadores, à deteção e reconhecimento dos objetos. O acesso às funções desta biblioteca é feito através da classe “CascadeClassifier”, que encapsula os métodos para parametrização da plataforma e

execução dos procedimentos para identificação de objetos em escala variável. O algoritmo da Figura 25 representa o processo implementado para o reconhecimento de “Buracos na estrada” numa fotografia:

```

1: //Algoritmo de detecção e classificação de objetos
2: Entrada: bitmapImagem, xmlClassificador, fatorEscala
3: Saída: contemObjeto
4: Início
5: //Carregar ficheiro XML que descreve o classificador na plataforma OpenCV
6: ClassificadorCascata <- CarregarClassificadorEmCascata(xmlClassificador);
7: //Converter a fotografia recebida por parâmetro numa representação matricial
8: //Conversão para o formato Alpha Red Green Blue (ARGB) 8888 (8 bits por cada canal). Cada pixel é descrito em 4 bytes.
9: BitmapARGB <- ConverterBitmapParaARGB(bitmapImagem);
10: //Conversão da imagem ARGB para uma representação matricial, para processamento das zonas de interesse
11: MatrizPixeis <- CalcularMatrizARGB(bitmapARGB);
12: //Determinar o número de escalas a aplicar à imagem com base no fator de escala indicado por parâmetro
13: NúmeroEscalas <- CalcularNúmeroEscalas(matrizPixeis,fatorEscala);
14: Para contador <-1 ATÉ contador == NúmeroEscalas Faz
15:     //Escalar a imagem
16:     MatrizEscalada <- EscalarMatrizImagem(MatrizPixeis,contador*fatorEscala);
17:     //Calcular a imagem integral (realça a parte relevante da imagem a considerar, para aumentar o desempenho do classificador)
18:     MatrizImagemIntegral <- CalcularMatrizImagemIntegral(MatrizEscalada);
19:     EtapasClassificador <- ObterEtapasClassificador(ClassificadorCascata);
20:     //Percorrer a matriz segundo a metodologia "Sliding Window". Leitura da esquerda para a direita, de cima para baixo
21:     Para cada zonaPixeisI EM MatrizImagemIntegral Faz
22:         //É necessário fazer passar cada zona de pixeis da matriz integral por todas as etapas do classificador em cascata
23:         Para cada etapaI EM EtapasClassificador Faz
24:             //Obter os filtros de cada etapa
25:             FiltrosEtapa <- ObterFiltrosEtapa(etapaI);
26:             ResultadoFiltros <- [];
27:             Para cada filtroI EM FiltrosEtapa Faz
28:                 //Acumular os resultados do filtro de cada etapa
29:                 ResultadoFiltros <- ResultadoFiltros + ExecutarFiltro(zonaPixeisI,filtroI)
30:             FimPara
31:             //Avaliar se o valor acumulado dos filtros na zona de pixeis é suficiente para ultrapassar o limiar definido no classificador
32:             Se ResultadoFiltros < ObterLimitePorEtapa(etapaI) Então
33:                 UltrapassouEtapas <- FALSO;
34:                 QuebrarCiclo;
35:             FimSe
36:             FimPara
37:             //Se a zona de pixeis foi validada em todas as etapas do classificador, reconhecer que a imagem contém
38:             //o objeto descrito pelo classificador
39:             Se UltrapassouEtapas == VERDADEIRO Então
40:                 contemObjeto <- VERDADEIRO;
41:             Senão
42:                 contemObjeto <- FALSO;
43:             FimSe
44:         FimPara
45:     Devolver contemObjeto;

```

Figura 25 – Algoritmo de detecção e classificação de objetos

Este excerto de código retrata o processamento executado sobre as fotografias anexas a relatórios de ocorrências da aplicação. O algoritmo começa por carregar o classificador em cascata de um tipo de objeto (por exemplo, “Buracos na Estrada”), e converter a imagem indicada por parâmetro no formato ARGB (*Alpha, Red, Green e Blue*, o formato compreendido pela plataforma OpenCV). É mantido um classificador por tipo de objeto para, por um lado incluir facilmente novos tipos de objetos de forma independente, e por outro otimizar a pesquisa do classificador por apenas um tipo de objeto na imagem. De seguida, é criada uma representação matricial da imagem para possibilitar a realização de operações matemáticas, assim como fragmentação e escalamento. Esta matriz é utilizada em primeiro lugar para determinar, em conjunto com o fator indicado por parâmetro, o número de escalas a efetuar sobre a imagem, com vista a albergar todos os casos possíveis de diferença de escala entre a imagem original e as imagens do conjunto de treino.

Descoberto o número de vezes que é necessário redimensionar, é iniciado um ciclo para efetuar o seu escalamento progressivo da imagem, onde é também calculada a sua representação integral e o número de etapas que compõem o classificador.

O processamento de cada zona de píxeis é efetuado sobre uma representação simplificada da imagem original, denominada de imagem integral [Viola & Jones,2001]. A seguinte fórmula calcula a soma de píxeis acima e para a esquerda da posição indicada por x, y :

$$\ddot{i}(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (3)$$

,onde $\ddot{i}(x, y)$ é a função que representa a imagem integral, $i(x, y)$ a função que descreve a imagem original e x e y as coordenadas dos píxeis das imagens.

Segundo o estudo descrito em [Viola & Jones,2001], cada zona de píxeis a processar pelas etapas do classificador pode conter um número de retângulos representativos de objetos maior do que o número de píxeis constantes nessa zona, o que invalida o seu processamento em tempo útil. Para dar resposta a este problema, [Viola & Jones,2001] incluíram uma variante própria do algoritmo de otimização *AdaBoost* denominado de *Gentle AdaBoost*, responsável pela seleção eficiente dos retângulos a considerar, considerando apenas os exemplos que melhor diferenciem os exemplos positivos e negativos [Viola & Jones, 2001].

Para a determinação destes retângulos de píxeis, é utilizada a seguinte fórmula:

$$h_j(x) = \begin{cases} 1 & ,se \quad p_j f_j(x) < p_j \theta_j \\ 0 & ,senão \end{cases} \quad (4)$$

,onde x é uma zona de píxeis da imagem (por exemplo 24x24 píxeis), $h_j(x)$ a função que descreve o classificador, f_j uma possível representação do objeto, θ_j o limiar de aceitação e p_j a paridade.

Descobertos os retângulos a considerar na análise da imagem integral, é necessário percorrer todas as etapas do classificador em questão, fazendo passar cada retângulo por todos os filtros de cada etapa, e assim calcular o valor acumulado de correspondência da zona a ser examinada.

Se o valor acumulado para uma determinada etapa não ultrapassar o limiar mínimo de aceitação do classificador, a zona da imagem a ser analisada é imediatamente rejeitada como possível candidata a conter o objeto. Por outro lado, caso o valor acumulado seja igual ou superior ao definido em todas as etapas, a zona é marcada como possível de conter o objeto, e a função termina com a devolução do valor "VERDADEIRO".

Apesar de todas as otimizações existentes nos algoritmos da plataforma OpenCV, a incerteza do tempo de execução dos classificadores para uma determinada imagem levou a tornar a invocação dos métodos desta biblioteca assíncrona, evitando desta forma o bloqueio da interface gráfica da aplicação (ver Figura 26):



Figura 26 – Aplicação móvel – Reconhecimento de objetos na imagem

Quando é anexa uma imagem a um relatório de ocorrência, uma barra vertical de cor verde é apresentada ao utilizador, efetuando um varrimento que dá a sensação do processamento contínuo da fotografia. Quando a invocação do algoritmo de classificação termina, a barra é removida da imagem e é apresentado o resultado do processo de deteção e classificação ao utilizador. Caso seja detetado algum objeto na imagem, é retornada a *tag* que define esse mesmo objeto (ver Figura 27):

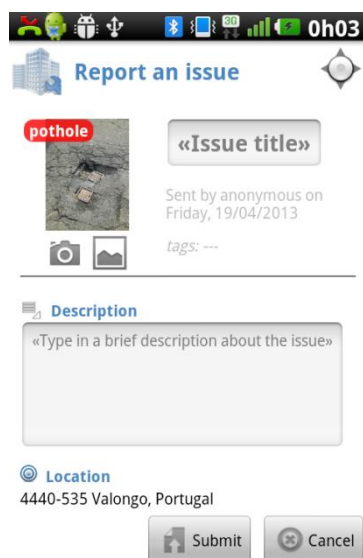


Figura 27 – Aplicação móvel – Atribuição de uma categoria (classificação)

5.5 Caso de estudo

O caso de estudo apresentado nesta secção procura corroborar a abordagem tomada nesta investigação, relativamente à implementação de um protótipo para a classificação automática de imagens em plataformas móveis. Para atingir este objetivo, é efetuada uma simulação com um conjunto de imagens de ocorrências reais do domínio “Buracos na Estrada”, retirando conclusões que permitem comprovar empiricamente o sucesso ou insucesso do algoritmo desenvolvido.

A simulação levada a cabo nesta investigação, considerou dois conjuntos de dados: um **conjunto de imagens para treino do classificador** “Buracos na Estrada”, e um **conjunto de imagens de teste do algoritmo**. Para garantir a maior fiabilidade possível, estes conjuntos são totalmente independentes, não partilhando qualquer informação entre si. As imagens para ambos os conjuntos foram obtidas de diversas fontes de acesso público na *Internet*, como os sítios *Web* de plataformas de notificação de problemas urbanos “SeeClickFix” e “FixMyStreet”.

O classificador de objetos foi desenvolvido com base na plataforma OpenCV, recorrendo aos utilitários *opencv_traincascade* (para criação de classificadores de objetos em cascata) e *opencv_createsamples* (para geração dos conjuntos de treino e de teste através da compilação das imagens originais). Do ponto de vista do algoritmo implementado, é apenas considerado o caso binário de deteção dos objetos: avalia se a imagem contém ou não o objeto, com base nas informações fornecidas pelo classificador. O algoritmo é, portanto, indiferente ao número de objetos que uma mesma imagem possa conter, retornando o valor positivo “um” caso exista pelo menos um objeto do tipo capturado pelo classificador, ou o valor “zero” caso não tenha sido detetado qualquer objeto. Para manter a coerência e facilitar a leitura e interpretação dos dados, o mesmo intervalo de valores (“um” e “zero”) foi considerado para a representação da presença ou ausência real de um objeto do tipo “Buracos na Estrada” nas imagens do conjunto de teste.

A análise estatística efetuada nesta secção permite não só determinar o grau de concordância do algoritmo com a realidade, como também retirar indicadores de desempenho e fiabilidade do sistema de reconhecimento e classificador desenvolvidos. No entanto, devido às dificuldades encontradas na recolha dos dados, e embora esta tenha sido feita de forma aleatória, o universo dos conjuntos de dados de treino e de teste é muito limitado, tanto em número de imagens, como na variabilidade de condições de luz e contraste das fotografias. Esta situação leva à construção de um classificador supervisionado especializado num certo domínio de imagens, não servindo como generalização de um qualquer tipo de objeto desse domínio (objetos com diferentes formas, diferentes contrastes na fotografia, etc.).

Não obstante, a elaboração deste protótipo permite validar o conceito idealizado, ao mesmo tempo que contribui na criação de uma base robusta para a adição futura de um número crescente de imagens, reaproveitando as teorias relacionadas com *crowd-sourcing* aplicadas no sistema.

5.5.1 Definição do Caso de Estudo

As imagens que formam o conjunto de dados do caso de estudo deste projeto foram obtidas através da descarga aleatória de fotografias anexas a relatórios de ocorrências, disponíveis para consulta pública nas principais plataformas de notificação de problemas urbanos.

O conjunto angariado conta com quarenta imagens do tipo “Buracos na Estrada”, e trinta imagens que não contêm o objeto a detetar, constituindo respetivamente exemplos positivos e negativos deste domínio. Deste conjunto, foram criados dois subgrupos de fotografias: um grupo de imagens para o treino do classificador, e outro grupo para o teste desse mesmo classificador. A Tabela 4 mostra a distribuição das várias amostras pelos grupos de treino e teste:

Tabela 4 — Subdivisão das imagens recolhidas

	Exemplos positivos (nº)	Exemplos negativos (nº)
Conjunto de treino	30	20
Conjunto de teste	10	10
Total amostrado	40	30

A subdivisão do conjunto de dados recolhidos nestes dois grupos a partir da mesma recolha de dados, deve-se à necessidade de imparcialidade e aleatoriedade na seleção dos exemplos que constituem cada um dos grupos. Desta forma, os testes são realizados sobre dados reais, estando sujeitos às mesmas condições utilizadas no treino do algoritmo. Nesta análise, foi utilizado o mesmo dispositivo para a realização de todos os testes: dispõe do sistema operativo *Android* versão 2.3.4, processador ARM V7 e biblioteca *OpenCV* 2.4.6.

Apesar de a amostra recolhida não ser representativa da generalidade dos casos de “Buracos na estrada” (muito devido à dificuldade dos processos de obtenção e tratamento das imagens), serve o propósito de validar o algoritmo desenvolvido, através da criação e alimentação de um classificador especializado num domínio restrito de imagens. Este classificador foi gerado com recurso aos programas *opencv_createsamples* e *opencv_traincascade*, que combinados possibilitam a definição de classificadores segundo um modelo em cascata. A Tabela 5 resume as parametrizações efetuadas, consistentes em ambas as ferramentas:

Tabela 5 — Parâmetros de configuração dos utilitários de criação do classificador

Parâmetro	Valor
Número de exemplos positivos utilizados	30
Número de exemplos negativos utilizados	20
Altura (em proporção) das imagens	40 Píxeis
Largura (proporção) das imagens	80 Píxeis
Técnica de aceleração (BOOST)	GAB ¹⁰
Número máximo de etapas do classificador	10
Tipo de identificação de objetos	Haar
Rácio de falso alarme	<=0.5
Rácio de correspondência	>=0.995

Na geração do classificador, foram utilizados grande parte dos exemplos positivos e negativos da amostragem realizada, reduzidos proporcionalmente (oitenta píxeis em largura, e quarenta píxeis em altura). Foi definido o limite de dez etapas do classificador (devido ao baixo número de exemplos positivos, o classificador atingiu o limite de rácio de falsos positivos à oitava etapa), e utilizada a técnica GAB para o rápido processamento das zonas de píxeis identificadas pelo método Haar. O rácio que indica a ocorrência de falsos alarmes em cada etapa do classificador foi definido para o máximo de 0,5, e o rácio de correspondência mínima em cada etapa para 0,995. Como em cada estágio da cascata ocorre uma multiplicação destes rácios pelo valor atingido na etapa, é possível determinar o rácio global do classificador para cada uma destas variáveis com a seguinte fórmula [OpenCV Dev Team, 2013]:

$$R_{Global} = R_{etapa}^n \quad (5)$$

,onde n é o número de etapas efetivas do classificador (por exemplo, 8) e R_{etapa} o valor do rácio por etapa a considerar. No caso concreto deste classificador, onde apenas foram consideradas oito etapas, o rácio global de falso alarme é de aproximadamente 0,0039, e o rácio global de correspondência atinge aproximadamente 0,9607 (ambos os valores foram arredondados à quarta casa decimal).

Em relação ao algoritmo implementado na aplicação móvel, foram igualmente realizadas parametrizações para oferecer uma melhor resposta aos classificadores gerados:

- **Transformação das imagens de teste:** as imagens submetidas na aplicação são convertidas para escala de cinzas;
- **Escala:** é criada uma pirâmide de fotografias de várias escalas, reduzindo a imagem original 10% em cada iteração do algoritmo;
- **Canny Pruning:** remoção inteligente de informação redundante nas imagens.

¹⁰ Gentle AdaBoost

5.5.2 Análise do Caso de Estudo

A confrontação dos resultados dos testes efetuados ao par algoritmo/classificador de “Buracos na estrada” com a realidade permitem determinar não só o grau de fiabilidade e concordância do sistema de reconhecimento desenvolvido, como também validar a abordagem tomada e sustentar a expansão do algoritmo a outros domínios. A Tabela 6 resume o resultado da análise do conjunto de teste:

Tabela 6 — Resultados do reconhecimento das imagens de teste

Imagem	Grupo	Largura (px)	Altura (px)	Tempo de execução (ms) ¹¹	Resultado esperado	Resultado do algoritmo
1	1	338	600	345	1	0
2	1	450	600	641	1	0
3	1	800	575	338	1	1
4	1	450	600	734	1	0
5	1	800	450	510	1	0
6	1	800	600	584	1	0
7	1	450	600	797	1	0
8	1	450	600	457	1	0
9	1	450	600	295	1	0
10	1	800	450	361	1	0
11	2	1600	1332	453	0	1
12	2	800	600	274	0	0
13	2	1600	1200	216	0	0
14	2	1024	682	364	0	0
15	2	1600	1200	238	0	0
16	2	3264	2448	153	0	0
17	2	1440	1080	681	0	0
18	2	1080	810	390	0	0
19	2	3888	2592	338	0	0
20	2	1024	768	221	0	0

As imagens do conjunto de teste foram subdivididas em dois grupos lógicos: imagens que contêm o objeto a identificar (exemplos positivos), e imagens que não contêm o objeto (exemplos negativos), respetivamente grupos 1 e 2. A discriminação da resolução da imagem nas suas componentes de altura e largura é necessária para justificar a variação do tempo de execução do algoritmo, uma vez que imagens maiores são sinónimo de mais iterações do algoritmo.

Os parâmetros “Resultado esperado” e “Resultado do algoritmo” indicam a presença (valor “1”) ou ausência (valor “0”) de pelo menos um objeto do tipo “Buracos na estrada” nas imagens de teste. No parâmetro “Resultado esperado”, a verificação da presença ou ausência

¹¹ O tempo de execução representa apenas o tempo gasto (milissegundos) no processamento das etapas do classificador, e não considera o processamento associado à ligação à plataforma OpenCV ou ao carregamento do ficheiro XML do classificador.

dos objetos foi feita manualmente, ao contrário do parâmetro “Resultado do algoritmo”, que espelha o sucesso ou insucesso do reconhecimento automático.

O gráfico apresentado de seguida (ver Figura 28) confronta os resultados obtidos pelo algoritmo com os valores esperados para cada imagem:

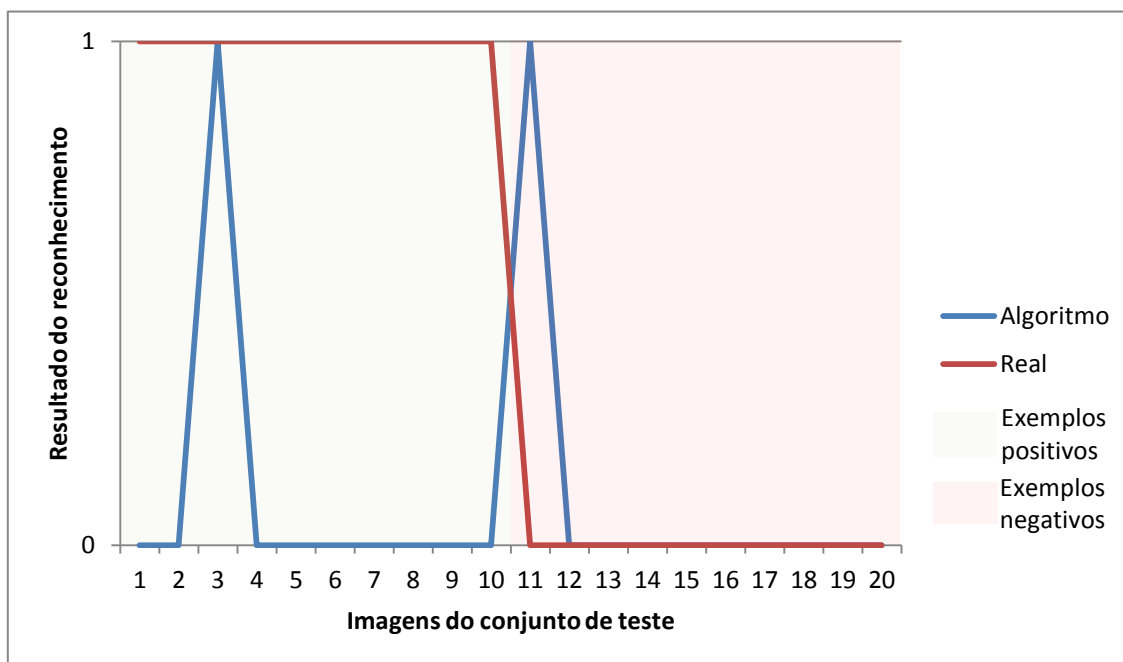


Figura 28 – Gráfico de comparação dos resultados do reconhecimento

Ao efetuar a leitura do gráfico, é possível identificar uma disparidade elevada dos resultados do algoritmo no grupo dos exemplos positivos de imagens, devolvendo apenas uma correspondência na imagem 3 (ver Anexo II) entre dez possíveis (representa uma eficácia de 10%). Por outro lado, no grupo de exemplos negativos, o algoritmo rejeitou a existência de objetos do tipo “Buracos na estrada” em nove de dez situações possíveis, o que resulta numa eficácia de 90% no reconhecimento. Estes índices de eficácia permitem concluir que o par algoritmo/classificador está mais suscetível à rejeição de novas imagens do que à sua aceitação, apontando para uma classificação pessimista.

Para enriquecer a análise efetuada, e fornecer uma estimativa indique a concordância do algoritmo desenvolvido com a realidade, foi calculada a correlação de *Pearson* entre os resultados obtidos pelo par algoritmo/classificador (X) e a realidade (Y). Esta correlação permite determinar o grau com que duas variáveis (por exemplo X e Y) se relacionam:

Tabela 7 — Resultados coincidentes segundo o coeficiente de correlação de Pearson

Imagem	Resultado esperado (Y)	Resultado do algoritmo (X)	Resultado coincidente
1	1	0	0
2	1	0	0
3	1	1	1
4	1	0	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	1	0	0
10	1	0	0
11	0	1	0
12	0	0	1
13	0	0	1
14	0	0	1
15	0	0	1
16	0	0	1
17	0	0	1
18	0	0	1
19	0	0	1
20	0	0	1

A Tabela 7 compara o resultado esperado (Y) com o resultado obtido pelo algoritmo (X) para todas as imagens do conjunto de teste, representando com o valor “1” a presença de um objeto do tipo “Buracos na estrada” e o valor “0” para a sua ausência. A coluna “Resultado coincidente” considera o valor “1” para a correspondência entre as variáveis X e Y, e o valor “0” para a sua discordância.

Com estes dados foi possível determinar o coeficiente de correlação de *Pearson* segundo a seguinte fórmula:

$$r_{xy} = \frac{S_{xy}}{S_x \cdot S_y} \quad (6)$$

,onde r_{xy} é o coeficiente de correlação para X e Y, S_{xy} representa a covariância e S_x e S_y os desvios-padrão.

Para as variáveis X e Y o valor do coeficiente de correlação de *Pearson* determinado é:

$$r_{xy} = -2,31296 \times 10^{-17} \quad (7)$$

Segundo Bryman & Cramer [Bryman & Cramer, 2003], a intensidade da correlação define-se através dos seguintes intervalos:

- Muito baixa, quando $r_{xy} < 0,20$;
- Baixa, quando $0,20 \leq r_{xy} < 0,40$;
- Moderada, quando $0,40 \leq r_{xy} < 0,70$;
- Alta, quando $0,70 \leq r_{xy} < 0,90$;
- Muito alta, quando $r_{xy} \geq 0,90$.

O sinal da correlação indica se estamos perante uma correlação positiva ($r_{xy} > 0$), ou uma correlação negativa ($r_{xy} < 0$) em que, caso uma das variáveis aumente, a outra diminui sempre. Neste caso de estudo, o valor obtido para r_{xy} foi de $-2,31296 \times 10^{-17}$, significando que estamos diante de uma **correlação negativa** (muito próxima de zero) de **intensidade muito baixa** entre as variáveis X e Y. Este valor justifica-se principalmente pela deficiência da deteção dos casos positivos (apenas uma em dez possíveis deteções), uma vez que, no processamento dos exemplos negativos, se verificou um alto grau de concordância (nove em dez possíveis) entre o algoritmo e a realidade. Em todo o caso, e muito embora o valor decorrente da análise da correlação de *Pearson* indique uma relação linear quase inexistente entre as variáveis, é possível concluir que o algoritmo teve uma eficácia no reconhecimento global (considerando tanto os exemplos positivos, como os exemplos negativos) de 50% (dez classificações corretas em vinte possíveis).

O cálculo do indicador de eficácia não permite categorizar as situações de erro do par classificador/algoritmo, sendo necessária a realização de uma análise de Falsos Positivos (FP) e Falsos Negativos (FN), identificados no conjunto de teste:

Tabela 8 — Falsos Positivos e Falsos Negativos

	Nº falsas deteções	Nº máximo de imagens	Percentagem de erro (%)
Grupo exemplos positivos	9	10	90
Grupo exemplos negativos	1	10	10
Total	10	20	50

A Tabela 8 resume o número de falsas deteções por grupo de imagens do conjunto de teste, fornecendo uma percentagem de erro global, e em cada um dos subconjuntos de exemplos positivos e negativos. A designação atribuída às falsas deteções está dependente do grupo de exemplos a que pertencem: uma classificação errada no grupo de exemplos positivos (não

reconhecimento da imagem) constitui um Falso Negativo (FN); uma classificação errada no grupo de exemplos negativos (reconhecimento da imagem) representa um Falso Positivo (FP).

O gráfico da Figura 29 ilustra a clara diferença do número de falsas deteções entre os grupos de imagens positivas e negativas:

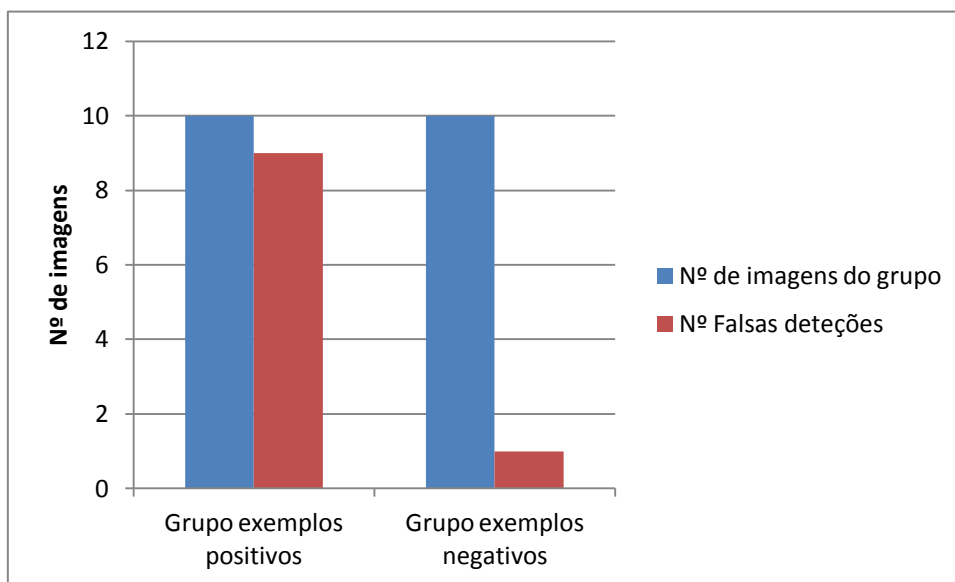


Figura 29 – Gráfico de comparação do número de falsos positivos e falsos negativos

Neste gráfico, verifica-se o predomínio de falsas deteções no grupo de exemplos positivos, representando nove situações FN em dez possíveis (90%). Por outro lado, no grupo de exemplos negativos apenas se verificou uma situação FP em dez possíveis, resultando numa taxa de erro de 10%. Estes dados indicam que o par algoritmo/classificador revela falta de sensibilidade no reconhecimento de novas imagens de “Buracos na estrada”, atingindo uma taxa de eficácia na deteção de somente 10%. Já no grupo de exemplos negativos, a taxa de ocorrência de FP ficou-se pelos 10%, apontando para uma classificação de imagens pessimista.

5.5.3 Conclusões

A análise estatística realizada nesta secção permitiu estabelecer métricas para a análise do desempenho e fiabilidade do algoritmo desenvolvido no reconhecimento de novas imagens do tipo “Buracos na estrada” submetidas pelos utilizadores. Para isso, foram compilados dois conjuntos de dados: um conjunto de imagens para treino do algoritmo, e um conjunto de dados de teste. O treino do classificador com base em imagens de “Buracos na estrada” existentes origina um modelo de classificação supervisionada, permitindo o reconhecimento de novas imagens que tenham características semelhantes ao domínio capturado nestes exemplos. O conjunto de teste constitui o objeto de estudo desta secção, sendo alvo de vários cálculos e comparações com o objetivo de determinar um grau indicador de confiança do algoritmo. Estes dois conjuntos foram ainda subdivididos em grupos que representam os exemplos positivos e negativos do domínio a reconhecer.

As dificuldades encontradas na recolha aleatória dos dados, limitaram o universo dos conjuntos de dados de treino e de teste, permitindo reunir apenas um pequeno grupo de imagens com baixas variações de forma, condições de luz e contraste, não servindo como generalização absoluta da classe “Buracos na estrada”. Esta recolha deficiente de exemplos do domínio em causa, principalmente de elementos positivos, justifica o baixo teor de reconhecimento de novas imagens. O estudo feito por [Lienhart et al.,2003], que visava o teste e a comparação de métodos de aceleração para algoritmos de classificação em cascata, refere a utilização de 5000 exemplos positivos e 3000 exemplos negativos como sendo uma quantidade aceitável de imagens para a geração de um classificador de reconhecimento de caras. Tendo estes valores como referência, e considerando que o domínio de “Buracos na estrada” é muito abrangente e sujeito a inúmeras variações de forma, luz e contraste, o número de imagens para a geração de um classificador deste tipo teria de conter no mínimo 5000 exemplos positivos e 3000 exemplos negativos.

O primeiro fator de comparação entre a classificação do algoritmo e a realidade procurou confrontar os resultados obtidos pelo algoritmo no processamento do conjunto de teste com o esperado. Esta confrontação resultou no cálculo dos valores de eficácia do algoritmo em cada um dos grupos de imagens do conjunto de teste: rácio de 10% na deteção de imagens novas imagens (atribuição da classificação “Buracos na estrada”), e rácio de 90% na rejeição de imagens que não contêm objetos do tipo “Buracos na estrada”. Os valores de eficácia obtidos em ambos os grupos permitem concluir que o algoritmo é resistente ao reconhecimento de novas imagens, indicando uma classificação do tipo pessimista. No entanto, em termos globais, o algoritmo foi eficaz em 50% das situações (dez imagens classificadas corretamente, entre vinte possíveis).

O segundo fator a ser avaliado foi a existência ou não de uma relação linear entre os resultados obtidos pelo classificador (variável X) e os resultados esperados (variável Y). Este estudo recorreu à correlação de *Pearson*, e permitiu determinar de que forma e relevância estas variáveis estão correlacionadas entre si. O coeficiente de *Pearson* calculado

($-2,31296 \times 10^{-17}$) sugere uma correlação negativa (muito próxima de zero) de intensidade muito baixa, o que significa que quando uma variável tende a aumentar, a outra tende a ter o comportamento inverso.

O último fator a ser considerado, permitiu a categorização das situações de erro ocorridas no reconhecimento das imagens do conjunto de teste, decompondo as falsas detecções em dois grupos: Falsos Positivos (FP) e Falsos Negativos (FN). Nesta análise verificou-se o predomínio de falsas detecções no grupo de exemplos positivos (erros do tipo FN), contrastando com o baixo teor de erros do tipo FP no grupo de exemplos negativos, completando a afirmação feita na avaliação do primeiro fator acerca do pessimismo do classificador.

O resultado desta análise mostra que o classificador gerado não representa a generalidade da classe de objetos “Buracos na estrada”, uma vez que o conjunto de treino reúne poucos exemplos positivos desta patologia de estrada, não considerando todas as variantes de forma, iluminação e contraste possíveis de encontrar nas fotografias submetidas pelos utilizadores. No entanto, o estudo efetuado comprovou o conceito idealizado para a notificação de problemas urbanos, contribuindo para a criação de uma base robusta para o futuro, onde bastará aumentar exponencialmente o conjunto de imagens de treino para obter resultados mais aproximados com a realidade. A capacidade de aprendizagem do algoritmo ficará a cargo dos seus utilizadores, uma vez que é a partir das suas submissões e respetivas e catalogações manuais que novas imagens poderão ser incluídas no conjunto de treino. O modelo de colaboração proposto assenta nos moldes de aplicações de *crowd-sourcing*, que delegam nos seus utilizadores a tarefa de alimentar e fazer prosperar os sistemas. Assim sendo, os utilizadores deste sistema desempenham um papel preponderante no desenvolvimento da plataforma, ao mesmo tempo que colaboram ativamente na resolução de problemas da sua comunidade local.

6 Conclusões

6.1 Resumo

O objetivo primordial desta investigação passou pela idealização, conceção e implementação de uma solução completa de notificação de problemas urbanos de carácter não urgente, procurando conferir aos cidadãos a capacidade de reportarem situações que perturbem o seu dia-a-dia diretamente às entidades municipais responsáveis.

A área de notificação de problemas urbanos tem vindo a sofrer alterações na sua forma de interagir com os cidadãos, necessitando de se adaptar às constantes transformações tanto do mercado como da sociedade civil. Numa era dominada pela computação móvel, o uso sistemático de um *smartphone* ou *tablet* para a execução das mais variadas tarefas abre horizontes para a temática de resolução de problemas, que até então não tinha uma resposta rápida e eficaz. Os sistemas de notificação clássicos, como linhas telefónicas e caixas de correio eletrónico, imputavam custos elevados às entidades municipais para efetuar a receção e triagem das ocorrências, atuando como meros mediadores entre os cidadãos que reportam situações, e as entidades responsáveis pela sua resolução.

Com os recentes avanços tecnológicos, surge espaço para sistemas de cariz cívico, em que os cidadãos podem, autonomamente, promover o crescimento da sua comunidade e notificar diretamente as entidades municipais, permitindo simultaneamente a poupança de recursos humanos e financeiros das entidades de gestão local. Esta oportunidade de negócio cativou a equipa do projeto para o desenvolvimento de um sistema deste tipo, impulsionando os estudos preliminares de análise da viabilidade do projeto. O *know-how* detido pela equipa do projeto na área de sistemas de informação geográfica, aliado à recente oportunidade de desenvolvimento de uma solução para a recolha de resíduos urbanos, reuniu as condições ideais para implementação desta plataforma de notificação de ocorrências, dando-se início aos trabalhos de investigação e desenvolvimento.

Neste sentido, foi realizado um levantamento do estado da arte da área de notificação de problemas urbanos, com vista a determinar as funcionalidades base da plataforma, assim

como identificar lacunas comuns a todos os sistemas que compõem a oferta do mercado. Deste estudo, resultou a identificação e seleção da demarcação manual e do reconhecimento de padrões em imagens como características inovadoras a abordar neste trabalho. Esta escolha sustentou-se na análise comparativa efetuada entre todas as ferramentas de notificação de problemas estudadas, cuja avaliação permitiu concluir que as principais falhas das aplicações centram-se nas dificuldades que impõem ao cidadão aquando da criação de um relatório de uma ocorrência, exigindo demasiado esforço para efetuar o seu preenchimento. Pretende-se, com a implementação destas duas funcionalidades, minimizar o tempo de preenchimento e processamento dos relatórios de ocorrências, através da atribuição de uma classificação automática com base nas imagens anexas, e a possibilidade de demarcar claramente a área problemática, respetivamente.

Para uma melhor compreensão das temáticas a abordar, foi elaborada uma revisão bibliográfica de cada uma, resumindo as diferentes aplicações, métodos e técnicas existentes em sistemas operativos móveis. Em relação à demarcação manual em imagem, a análise levada a cabo concluiu que a técnica a implementar assenta nas bibliotecas de gráficos 2D do sistema operativo *Android*, utilizando as APIs de baixo nível para captar os movimentos de delimitação do utilizador e efetuar o desenho das linhas no ecrã. No caso do reconhecimento de padrões em imagem, foi acordada a integração da tecnologia OpenCV no projeto, possibilitando a execução de classificadores em cascata sobre as fotografias anexas aos relatórios de ocorrências.

Finda a fase de estudo e levantamento do estado da arte das várias áreas, foi iniciado o desenvolvimento da solução, compreendendo várias etapas, como a estruturação do modelo proposto, a modelação de persistência da informação, ou a conceção da aplicação móvel. O foco desta investigação recaiu sobre esta última componente, da qual fazem parte os módulos de demarcação e reconhecimento de imagens, que representam o carácter inovador deste trabalho. No módulo de demarcação manual de objetos em imagens, foi construído um controlo de interface gráfica, capaz de captar os movimentos efetuados pelo utilizador sobre o ecrã do dispositivo. De seguida, o controlo encarrega-se de desenhar o caminho traçado sobre a imagem, utilizando linhas com transições curvas suaves entre os segmentos de reta, numa tentativa de imitar o comportamento típico do desenho manual com lápis. O módulo de reconhecimento de imagem é o mais complexo de toda a solução, correspondendo a grande parte do esforço realizado nesta investigação. Para o desenvolvimento deste protótipo, foi apenas gerado um classificador em cascata para objetos do tipo “Buracos na estrada”, com o propósito de validar a abordagem tomada. O módulo de reconhecimento desenvolvido é responsável por carregar o ficheiro que descreve o classificador, e percorrer as várias zonas da imagem, executando sequencialmente as etapas e filtros definidos no classificador. Caso uma zona da imagem ultrapasse todas as etapas do classificador, é marcada como uma possível correspondência do objeto a identificar, desencadeando o processo de atribuição de uma *tag* ao relatório.

Por fim, o par algoritmo/classificador desenvolvido é avaliado, com recurso a um conjunto de imagens de teste obtidas a partir de plataformas públicas de notificação de problemas.

6.2 Objetivos Alcançados

Na fase inicial desta investigação, foram realizados diversos estudos com vista a suportar as decisões tecnológicas do projeto, através da comparação de aplicações e tecnologias preponderantes nas várias temáticas abordadas.

Em primeiro lugar, foi efetuado um levantamento do estado da arte das aplicações de notificação de problemas urbanos, a fim de determinar funcionalidades típicas que os utilizadores esperam deste tipo de aplicações, e também identificar as lacunas comuns a todos os sistemas. Este estudo condicionou toda a investigação consequente, uma vez que permitiu idealizar as características inovadoras do produto a desenvolver, e assim definir uma linha de pensamento diferenciadora da concorrência.

Após a identificação da demarcação manual em imagens e do reconhecimento de padrões em imagem como pontos-chave a desenvolver nesta investigação, foi necessário efetuar uma revisão bibliográfica sobre cada uma destas áreas, para compreender a sua extensão, complexidade e viabilidade de integração no projeto. As revisões do estado da arte realizadas, permitiram determinar não só conjunto de aplicações, tecnologias e técnicas relevantes que constituem a oferta atual do mercado, mas também auxiliar a tomada de decisão de quais mecanismos de anotação e reconhecimento de imagem integrar na solução.

O principal objetivo deste trabalho focou a idealização, conceção, especificação e implementação de uma plataforma de notificação de problemas urbanos, distinta da oferta atual do mercado pelo modo facilitado de notificar as ocorrências. Para o atingir, foi construída uma arquitetura assente numa estrutura modular, que tem por base um portal com capacidades de gestão de informação georreferenciada desenvolvido pela equipa do projeto. O sistema de informação geográfico que está na base da solução é apresentado de um modo sucinto neste documento, elucidando o papel que os vários módulos e aplicações setoriais, que o constituem, desempenham. É também proposto um modelo para dar resposta aos requisitos funcionais e não funcionais, extraídos a partir da observação e interação com utilizadores reais.

Posteriormente, o modelo proposto para a solução guiou os trabalhos de implementação da plataforma, onde foi necessário identificar as tecnologias a integrar no projeto, conceber um modelo de persistência da informação, estruturar a camada de serviços e construir a aplicação móvel. Nesta fase, os desenvolvimentos da camada de serviços e da aplicação móvel eram feitos de forma concorrente, para não estagnar a evolução da aplicação móvel. Neste campo, a complexidade da aplicação móvel, muito devido à inclusão de técnicas de demarcação e reconhecimento em imagem, levou à sua subdivisão em módulos, permitindo a realização de tarefas focadas em objetivos concretos.

Por último, foram realizados testes para validação dos algoritmos de reconhecimento e classificação desenvolvidos, com o objetivo de corroborar a abordagem tomada nesta investigação.

6.3 Limitações e Trabalho Futuro

Todos os ramos da ciência têm margem de progressão nas atividades que desenvolvem, estando sujeitas ao aparecimento de novas soluções, metodologias e técnicas que promovem a aquisição de novos conhecimentos capazes de questionar o que até então eram factos comprovados.

As áreas de notificação de problemas urbanos, anotação e reconhecimento de padrões em imagem, assim como o trabalho desenvolvido nesta investigação, não são uma exceção à regra, estando naturalmente suscetíveis a atualizações tecnológicas e melhoramentos nos seus procedimentos. Nesta secção, são capturadas as limitações encontradas na solução atual, perspetivando desenvolvimentos futuros para a sua resolução. São ainda apresentadas algumas ideias para complementar a oferta disponibilizada, tendo em vista a sua adaptação a novas realidades.

A principal limitação da solução proposta nesta investigação reside na validação incompleta do algoritmo de classificação de objetos do tipo “Buracos na estrada”. Apesar da possibilidade de implementação de um classificador de imagens em ambiente móvel ter sido corroborada, não se obtiveram bons resultados na fase de testes e validação do algoritmo desenvolvido. Esta situação deve-se ao facto de não existirem bases de dados de imagens dedicadas a estas patologias, sendo necessário proceder à sua recolha e tratamento manuais. As dificuldades nos processos de recolha e tratamento das imagens, associadas à falta de diversidade das fontes de informação (imagens com variações relevantes nas condições de luminosidade, contraste e ruído), contribuíram para a geração de um conjunto de treino especializado num conjunto finito de possibilidades, não servindo como generalização da classe de objetos “Buracos na estrada”. Embora a recolha de amostras em massa estivesse fora do âmbito desta investigação, está prevista a elaboração de um estudo focado na determinação e colheita do número de amostras necessário para a generalização de várias classes de objetos, como “Buracos na estrada”, grafitis, iluminação pública danificada, entre outras.

A ausência de um formato padrão para a representação de anotações nas imagens é outra das limitações deste projeto. A solução adotada passa pela geração de novas imagens, sobrepondo as anotações nas fotografias originais, impossibilitando posteriormente a sua divisão. Esta abordagem, apesar de funcional, impede que outras aplicações consigam eliminar ou esconder as anotações de forma simples. Para solucionar este problema, está prevista a definição de uma proposta para um formato padrão de anotações em imagem, garantindo a interoperabilidade com outras aplicações.

Por último, seria interessante disponibilizar a aplicação desenvolvida noutros sistemas operativos móveis como *iOS* e *Windows Phone*, alargando ainda mais o leque de potenciais utilizadores do sistema.

6.4 Considerações Finais

Em primeiro lugar, é importante referir o quão gratificante e prazeroso foi a realização de todo este trabalho de investigação, não só pelos conhecimentos adquiridos sobre as várias temáticas abordadas, como pela experiência profissional ganha em novos domínios do saber.

Com grande satisfação considera-se que o trabalho desenvolvido contribui para o avanço na área das aplicações de notificação de problemas urbanos, abrindo novos horizontes e desafiando, em matéria de inovação, os sistemas existentes. Esta situação só foi possível graças à colaboração de figuras realmente extraordinárias nas suas áreas de conhecimento, cujos esforços conjuntos resultaram num produto de alta qualidade.

Para finalizar, gostaria de salientar que o empenho e dedicação a este projeto permitiram a conquista de valências muito importantes num futuro próximo, uma vez que este terá continuidade e aplicabilidade em contextos reais.

Referências

- [Agência para a Modernização Administrativa, 2008] Agência para a Modernização Administrativa, IP, 2008 “Modernização administrativa – A minha rua”, Sítio Rede Comum de Conhecimento: A minha Rua <http://www.rcc.gov.pt/Directorio/Temas/MA/Paginas/A-Minha-Rua.aspx> [Último acesso: 29 de Julho de 2013]
- [Agência para a Modernização Administrativa, 2009a] Agência para a Modernização Administrativa, IP, 2009 “A minha rua – Reportar ocorrência”, Sítio Portal do Cidadão: A minha Rua <http://www.portaldocidadao.pt/portal/aminharua/situationReport.aspx> [Último acesso: 29 de Julho de 2013]
- [Agência para a Modernização Administrativa, 2009b] Agência para a Modernização Administrativa, IP, 2009 “Programa 2009 – Medidas em destaque”, Sítio Simplex: Quanto mais simples, melhor: http://www.simplex.pt/2009programa/programa2009_05MedidasEmDestaque.html [Último acesso: 29 de Julho de 2013]
- [Agência para a Modernização Administrativa, 2013] AMA - Agência para a Modernização Administrativa, IP datad , “Informação Geral: Sobre o Portal”, Sítio Portal do Cidadão: <http://www.portaldocidadao.pt/PORTAL/pt/informacao+geral/Sobre> [Último acesso: 30 de Julho de 2013]
- [Ari & Aksoy, 2010] Ari C.; Aksoy S., "Unsupervised classification of remotely sensed images using Gaussian mixture models and particle swarm optimization," Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International , vol., no., pp.1859,1862, 25-30 Julho 2010
- [Bryman & Cramer, 2003] Bryman, Cramer, “Análise de dados em ciências sociais” 3ª ed, Celta Editora, Oeiras, 2003.
- [Byte Experts, 2013] Byte Experts, 2013, “Image Editor Android App”, Sítio Byte Experts: http://byteexperts.com/product_info.php?cPath=22&products_id=29?osCsId=4jp2iusmv63nljdm8s6bjo9qg3 [Último acesso: 22 de Agosto de 2013]
- [Camellia, 2013] Camellia, data “Camellia – Image Processing & Computer Vision library”, Sítio Camellia – Sourceforge: <http://camellia.sourceforge.net/> [Último acesso: 12 Setembro 2013]
- [Casanova et al., 2013] Casanova C., Franco A., Lumini A., Maio D., “SmartVisionApp: A framework for computer vision applications on mobile devices”, Expert Systems with Applications, Volume 40, Issue 15, Pages 5884-5894, 1 Novembro 2013,
- [Chandler, 2012] Chandler N. 2012, “What is Google Goggles?”, Sítio HowStuffWorks: <http://electronics.howstuffworks.com/gadgets/other-gadgets/google-goggles1.htm> [Último acesso: 8 Setembro 2013]
- [Chaovalit & Zhou, 2005] Chaovalit P., Zhou L. “Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches” - HICSS'05 Vol. 4. IEEE Computer Society, Washington, DC, USA, 2005.
- [Chen et al., 2012] Chen J., Zhu Y., Wang H., Jin W., Yu Y., 2012, “Effective and Efficient Multi-Facet Web Image Annotation”, Journal of Computer Science and Technology, Volume 27, Issue 3, 1 Janeiro 2012
- [Chuttur, 2011] Chuttur M. Y., “Defining and creating metadata for digital resources”, School of Library and Information Science, Indiana University, Library Student Journal, Março 2011
- [CitySourced, 2009] CitySourced 2009, CitySourced Sítio CitySourced: <http://www.citysourced.com> [Último acesso: 21 de Fevereiro de 2013]
- [Civic Apps, 2010] Civic Apps – For greater Portland 2010, “Contribute-Apps-PDX Reporter”, Sítio: <http://civicapps.org/apps/pdx-reporter> [Último acesso: 30 de Julho de 2013]
- [CivicTicket, 2012] CivicTicket 2012, “About Us – What is CivicTicket?”, Sítio CivicTicket:

- [Evernote Corporation, 2013] <http://www.civicticket.com> [Último acesso: 30 de Julho de 2013]
Evernote Corporation, 2013, "Skitch", Sítio Evernote:
<http://evernote.com/skitch/> [Último acesso: 22 de Agosto de 2013]
- [Foth et al., 2011] Foth M., Schroeter R., Anastasiu I., "Fixing the City One Photo at a Time: Mobile Logging of Maintenance Requests", Proceeding OzCHI '11 Proceedings of the 23rd Australian Computer-Human Interaction Conference Pages 126-129, 2011
- [Fraunhofer FOKUS, 2013] Fraunhofer FOKUS 2013, FixMyCity website: <http://www.fixmycity.de/> [Último acesso: 16 de Fevereiro de 2013]
- [Gan et al., 2012] Gan H., Sang N., Huang R., Tong X., Dan Z., "Using clustering analysis to improve semi-supervised classification", Neurocomputing, Volume 101, Pages 290-298, ISSN 0925-2012, 4 Fevereiro 2013
- [Google, 2012] Google 2012, "Visidon AppLock", Sítio Google Play: https://play.google.com/store/apps/details?id=visidon.AppLock&hl=pt_PT. Último acesso a 10 Setembro 2013..
- [Google, 2013a] Google, 2013, "Canvas and Drawables", Sítio Google Developer: <http://developer.android.com/guide/topics/graphics/2d-graphics.html> [Último acesso: 24 de Agosto de 2013]
- [Google, 2013b] Google 2013, "Editor de imagem", Sítio Google Play: <https://play.google.com/store/apps/details?id=com.pcvirt.ImageEditor>. Último acesso a 22 de Agosto de 2013.
- [Google, 2013c] Google 2013, "Skitch", Sítio Google Play: <https://play.google.com/store/apps/details?id=com.evernote.skitch> [Último acesso: 22 de Agosto de 2013]
- [Handa, 2012] Handa S., 2012, "Google Goggles", Department of Electrical And Computer Engineering ECE ILLINOIS, Google Goggles Presentation.
- [Howe, 2006] Howe J., 2006 "The Rise of Crowdsourcing", Sítio Wired: <http://www.wired.com/wired/archive/14.06/crowds.html> [Último acesso: 30 de Julho de 2013]
- [Hu & Lam, 2013] Hu J., Lam K., 2013, "An efficient two-stage framework for image annotation", Pattern Recognition, Volume 46, Issue 3, Março 2013, Pages 936-947, ISSN 0031-3203.
- [Isenhour, 2003] Isenhour, P. 2003, "Compressing Data Sent Over a Socket, Java Techniques website: <http://javatechniques.com/blog/compressing-data-sent-over-a-socket/> [Último acesso: 10 de Abril de 2013].
- [Itseez, 2013a] Itseez 2013, "OpenCV – ABOUT", Sítio OpenCV: <http://opencv.org/about.html>. [Último acesso: 10 de Setembro de 2013].
- [Itseez, 2013b] Itseez 2013, "OpenCV Wiki", Sítio Code - OpenCV <http://code.opencv.org/projects/opencv/wiki> Último acesso a 10 Setembro de 2013.
- [Jain et al., 2000] Jain, A.K.; Duin, R. P W; Jianchang Mao, "Statistical pattern recognition: a review," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.22, no.1, pp.4,37, Janeiro 2000
- [Jhet, 2012] Jhet C. W. 2012, "CivicTicket", Sítio Asian Social Inovation Camp: <http://sicampasia.com/kl-2012-2/meet-the-top-6/civic-ticket/> [Último acesso: 30 de Julho de 2013]
- [Koch & Brilakis, 2011] Koch, C., Brilakis I., 2011. "Pothole detection in asphalt pavement images", Advanced Engineering Informatics, Volume 25, Issue 3, Pages 507–515, Agosto 2011
- [Kooaba, 2011a] Kooaba, 2011, "Kooaba", Sítio Kooaba: <http://www.kooaba.com/en/home> [Último acesso: 13 Setembro de 2013]
- [Kooaba, 2011b] Kooaba 2011, "Kooaba - Pricing", Sítio Kooaba: http://www.kooaba.com/en/plans_and_pricing [Último acesso: 13 Setembro de 2013]

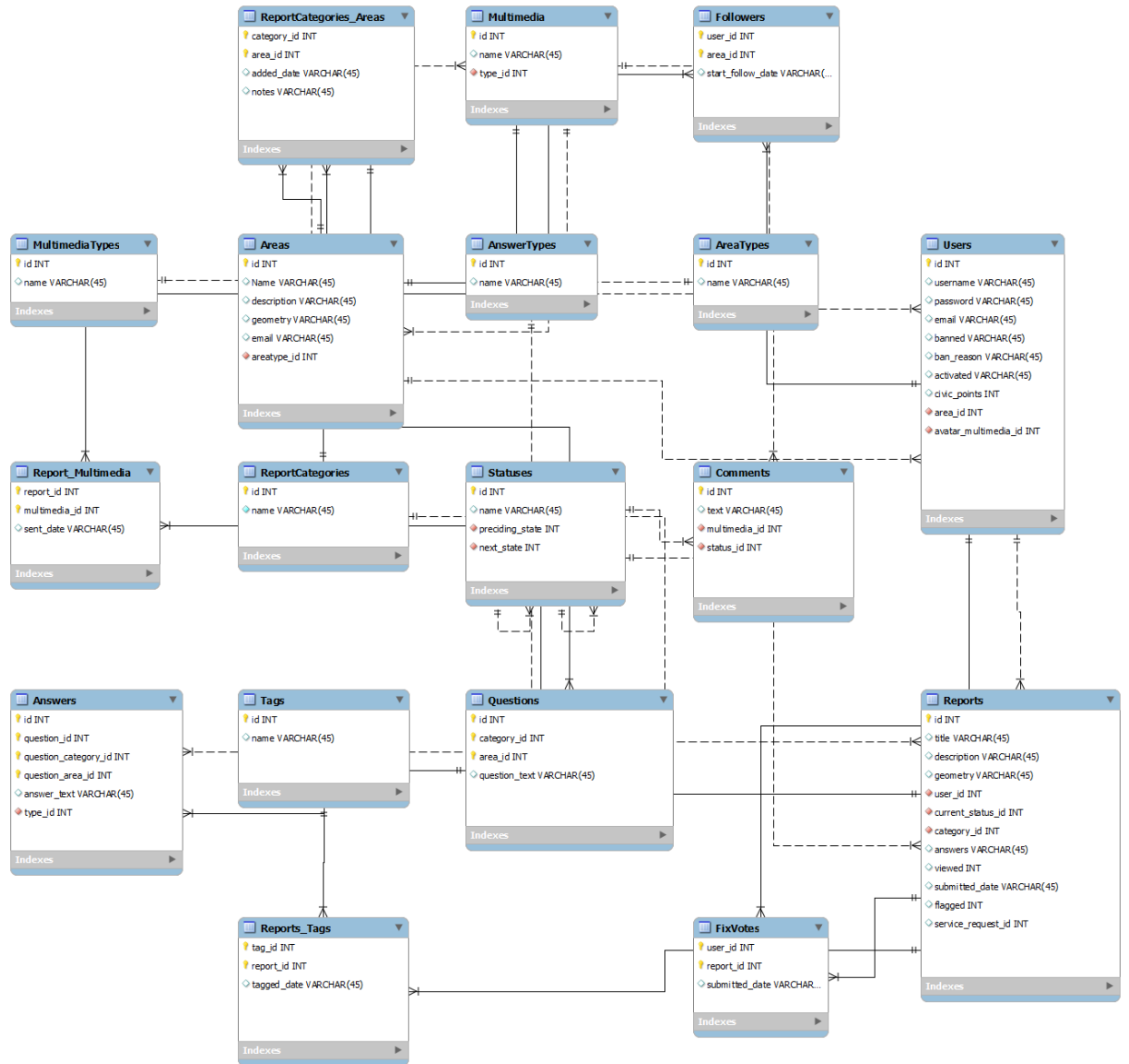
- [LEAD, 2013a] LEAD Technologies 2013, "About LEAD Technologies, Inc.", Sítio LEADTOOLS: <http://www.leadtools.com/corporate/corporate.htm> [Último acesso: 23 de Agosto de 2013]
- [LEAD, 2013b] LEAD Technologies datab, "LEADTOOLS Image Annotation SDK", Sítio LEADTOOLS: <http://www.leadtools.com/sdk/annotation/default.htm>. Último acesso a 23 de Agosto de 2013.
- [LEAD, 2013c] LEAD Technologies 2013, "LEADTOOLS Image Annotation SDK Objects", Sítio LEADTOOLS: <http://www.leadtools.com/sdk/annotation/objects.htm> [Último acesso: 23 de Agosto de 2013]
- [Lienhart et al., 2003] Lienhart, R., Kuranov, A., Pisarevsky, V, "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection," Pattern Recognition book., pp. 297-304, 2003.
- [Liu et al., 2006] Liu J., Sun J., Wang S. "Pattern Recognition: An overview" IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.6, June 2006
- [Lusa, 2013] Lusa, Agência, 2013 "Governo vai acelerar cortes nas despesas públicas", Sítio Visão: <http://visao.sapo.pt/governo-vai-acelerar-cortes-nas-despesas-publicas=f722630> [Último acesso: 9 de Abril de 2013]
- [Melbourne Design Awards, 2013] Melbourne Design Awards 2013, "2013 Melbourne Design Awards", Sítio: Melbourne Design Awards: http://melbournedesignawards.com.au/mda2013/entry_details.asp?ID=11767&Category_ID=5172 [Último acesso: 30 de Julho de 2013]
- [Moodstocks, 2013a] Moodstocks 2013, "Moodstocks", Sítio Moodstocks: <http://www.moodstocks.com/>. [Último acesso: 13 Setembro de 2013]
- [Moodstocks, 2013b] Moodstocks 2013, "Moodstocks - Features", Sítio Moodstocks: <http://www.moodstocks.com/features/> [Último acesso: 13 Setembro de 2013]
- [mySociety, 2013] mySociety 2013, FixMyStreet website: <http://www.fixmystreet.com/> [Último acesso: 14 Fevereiro de 2013]
- [Neuroph, 2013a] Neuroph 2013, "Neuroph", Sítio Neuroph/Sourceforge: <http://neuroph.sourceforge.net/index.html> [Último acesso: 13 Setembro de 2013]
- [Neuroph, 2013b] Neuroph 2013, "Neuroph – About", Sítio Neuroph/Sourceforge: http://neuroph.sourceforge.net/about_project.html [Último acesso: 13 Setembro de 2013]
- [Neuroph, 2013c] Neuroph 2013, "Neuroph – FAQ", Sítio Neuroph/Sourceforge: <http://neuroph.sourceforge.net/faq.html> [Último acesso: 13 Setembro de 2013]
- [Nvidia, 2013] Nvidia 2013, Sítio Tegra Android Toolkit documentation: http://docs.nvidia.com/tegra/data/How_to_Use_OpenCV_for_Tegra.html [Último acesso: 10 Abril de 2013]
- [Oliveira et al., 2012] Oliveira, L.; Rocha, A.; Coelho, A.; Dias, L.; Rodrigues, A.; Sousa, M.; Silva, D., "Implementing a regional spatial data infrastructure based on free software," Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on , vol., no., pp.1,5, 20-23 Junho 2012
- [Open311, 2013a] Open311, 2013, Sítio Open311 initiative: <http://open311.org/learn/> [Último acesso: 21 Fevereiro de 2013]
- [Open311, 2013b] Open311 – Wiki datab, "Supported Software", Sítio Open311: <http://wiki.open311.org/Software> [Último acesso: 30 Julho de 2013]
- [OpenCv Dev Team, 2013] OpenCv Dev Team 2013, "Cascade Classifier Training", Sítio OpenCV Docs: http://docs.opencv.org/doc/user_guide/ug_traincascade.html [Último acesso: 24 Setembro de 2013]
- [Outware Mobile, 2013a] Outware Mobile, 2013, "Description", Sítio Google Play:<https://play.google.com/store/apps/details?id=com.outware.snapsendsolve> [Último acesso: 30 Julho de 2013]
- [Outware Mobile, 2013b] Outware Mobile 2013, Sítio Outware Mobile: <http://www.outware.com.au/>

- 2013b] [Último acesso: 30 Julho de 2013]
- [Outware Mobile, 2013c] Outware Mobile 2013, "Our Apps", Sítio Outware Mobile: <http://www.outware.com.au/our-apps/> [Último acesso: 30 Julho de 2013]
- [Outware Mobile, 2013d] Outware Mobile 2013, "Snap Send Solve Version 2 goes live", Sítio: <http://www.outware.com.au/snap-send-solve-version-2-goes-live/> [Último acesso: 30 Julho de 2013]
- [Qualcomm, 2011a] Qualcomm, 2011 "Introducing FastCV: computer vision technology, tuned for mobile", Sítio Qualcomm: <http://www.qualcomm.com/media/blog/2011/10/25/introducing-fastcv-computer-vision-technology-tuned-mobile> [Último acesso: 12 Setembro de 2013]
- [Qualcomm, 2011b] Qualcomm 2011, "FastCV Getting Started Guide", Sítio Developer Qualcomm: <https://developer.qualcomm.com/mobile-development/mobile-technologies/computer-vision-fastcv/getting-started-guide> [Último acesso: 12 Setembro de 2013]
- [Qualcomm, 2011c] Qualcomm 2011, "FastCV Public API Documentation", Sítio Developer Qualcomm, <https://developer.qualcomm.com/docs/fastcv/api/index.html> [Último acesso: 12 Setembro de 2013]
- [Reed, 2013] Reed B., 2013 "Android's steady march to 1 billion activations gets visualized", BGR website: <http://bgr.com/2013/03/13/android-activation-growth-analysis-373572/> [Último acesso: 2 Abril de 2013]
- [RGK Center, 2012] RGK Center for Philanthropy and Community Service 2012, "CivicTicket - Make a civic-aware society!", Sítio Dell Social Innovation Challenge: <http://www.dellchallenge.org/projects/civicticket> [Último acesso: 30 de Julho de 2013]
- [Samsung, 2012] Samsung 2012, "Technical Docs - 2D Graphics in Android", Sítio Samsung Developers: <http://developer.samsung.com/android/technical-docs/2D-Graphics-in-Android> [Último acesso: 24 de Agosto de 2013]
- [Santos et al., 2013] Santos J., Rodrigues F., Oliveira L., "A Web & Mobile City Maintenance Reporting Solution". Procedia Technology, Paper accepted at CENTERIS 2013.
- [SeeClickFix, 2013] SeeClickFix 2013, SeeClickFix website: <http://seeclickfix.com/> [Último acesso: 8 de Março de 2013]
- [Shih et al., 2012] Shih C., Chu H., Chen Y., Wen C., 2012. "The effectiveness of image features based on fractal image coding for image annotation", Expert Systems with Applications, Volume 39, Issue 17, Pages 12897-12904, ISSN 0957-4174, 1 December 2012
- [SQLite, 2013] SQLite 2013, "SQLite - Home", Sítio SQLite: <http://www.sqlite.org/> [Último acesso: 18 de Setembro de 2013]
- [Steinberg, 2011] Steinberg, T. 2011, "FixMyStreet in Norway", mySociety Blog: <http://www.mysociety.org/2011/03/07/fixmystreet-in-norway/> [Último acesso: 14 de Fevereiro de 2013]
- [Takahashi, 2010] Takahashi D., 2010, "How Google Goggles works to deliver visual search results for mobile phones", Sítio Venture Beat: <http://venturebeat.com/2010/08/23/how-google-goggles-works-to-deliver-visual-search-results-for-mobile-phones/#eJXdQjkLOqGL28OU.99> [Último acesso: 5 Setembro 2013]
- [Tamilin et al., 2012] Tamilin, A, Carreras I., Ssebagala E, Opira A., Conci N., 2012. "Context-aware mobile crowdsourcing", In Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12) ACM, New York, NY, USA, 717-720, 2012
- [The City of Portland, 2013] The City of Portland – Oregon 2013, "PDX Reporter App", Sítio: <http://www.portlandoregon.gov/transportation/article/405043> [Último acesso: 30 de Julho de 2013]
- [Viola & Jones, 2001] Viola, P.; Jones, M., "Rapid object detection using a boosted cascade of

- simple features," Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on , vol.1, no., pp.1-511,1-518 vol.1, 2001
- [Visidon, 2013a] Visidon Ltd 2013, "Visidon AppLock info" Sítio Visidon: http://www.visidon.fi/en/Visidon_AppLock_info%23Table_of_contents [Último acesso: 9 de Setembro de 2013]
- [Visidon, 2013b] Visidon Ltd datab, "Visidon Products – Visidon AppLock for Android" Sítio Visidon: <http://www.visidon.fi/en/Products> [Último acesso: 9 de Setembro de 2013]
- [Watanabe, 1985] Watanabe, S. Pattern Recognition: Human and Mechanical. John Wiley & Sons, Inc., New York, NY, USA. 1985.

Anexo I

Modelo de dados de suporte à aplicação desenvolvida:



Anexo II

Alguns exemplos positivos de imagens utilizadas no conjunto de teste:

Imagem 1



Imagem 3



Imagem 5



Anexo III

Alguns exemplos positivos utilizados no conjunto de treino do classificador:

Imagem 1



Imagem 3



Imagem 12

