

Mestrado em Engenharia Informática

Dissertação/Estágio

Relatório Final

Desenvolvimento de uma Aplicação Móvel para o Turismo

Flávio da Silva Pereira

fpereira@student.dei.uc.pt

Orientadores:

Eng. António Pinho (Ubiwhere)

Prof. Dr. Daniel Castro Silva (DEI)

3 de Julho de 2013



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

Resumo

A crescente utilização dos *smartphones* veio revolucionar a maneira como as pessoas vivem atualmente. Esta área tecnológica está cada vez mais acessível ao cidadão comum, permitindo a utilização de aplicações móveis nas mais diversas situações. Neste estágio foi desenvolvida uma aplicação móvel, para *smartphone*, cujo objetivo é servir de guia turístico no interior do Jardim Botânico da cidade de Coimbra. A aplicação foi desenvolvida para o sistema operativo Android. Utilizando a aplicação, o visitante pode usufruir de uma experiência melhor e mais agradável durante a visita ao Jardim. A aplicação permite ao visitante visualizar dados sobre plantas presentes no interior do Jardim. Para além da visualização de informação em duas dimensões, o visitante pode escolher efetuar a visita através de um módulo de realidade aumentada, tornando a visita ainda mais agradável. Uma vez que as redes sociais estão cada vez mais presentes no quotidiano das pessoas, o visitante tem a opção de partilhar a sua experiência nas redes sociais Facebook e Twitter.

De modo a complementar a aplicação móvel, foi desenvolvida uma aplicação *web* para visualização de estatísticas. A aplicação móvel recolhe dados durante a visita, sendo estes enviados para o servidor. A aplicação para visualização de estatísticas vai buscar os dados, efetua tratamento sobre os mesmos e apresenta os resultados, num browser, recorrendo a um mapa. Com isto consegue-se saber, por exemplo, as plantas mais ou menos visualizadas no Jardim Botânico.

Palavras-chave

Aplicação móvel, Android, Serviços Baseados na Localização, *Context-Aware*, *Location-Aware*, Realidade Aumentada, Turismo.

Agradecimentos

Gostaria de agradecer em primeiro lugar aos meus pais e a minha irmã, que me apoiaram durante todo o meu percurso académico. Sem eles não teria chegado até aqui.

Um agradecimento ao meu Orientador do DEI, o Doutor Daniel Castro Silva, que me apoiou e orientou durante todo o estágio, mostrando sempre grande disponibilidade.

Gostaria também de agradecer à Ubiwhere que me acolheu durante este ano e ofereceu todas as condições necessárias para a concretização deste projeto. Um agradecimento ao meu orientador da empresa, o Engenheiro António Pinho, pelo apoio que me deu durante a execução deste projeto.

Finalmente gostaria de agradecer a todos meus amigos e à minha namorada.

Índice

1. Introdução	13
1.1 Enquadramento	13
1.2 Objetivos	14
1.3 Motivação	14
1.4 Estrutura do documento.....	15
2. Estado da Arte	16
2.1 Plataformas Móveis	16
2.2 Guias turísticos móveis baseados na localização	18
2.2.1 Soluções Analisadas	18
2.2.3 Revisão de guias existentes.....	24
2.3 Caracterização dos serviços baseados na localização	26
2.3.1: componentes principais:.....	26
2.3.2 Fatores chave para o sucesso dos SBL.....	28
2.3.3 Principais preocupações na implementação de SBL	29
2.4 Principais soluções para utilização de mapas	29
Soluções de mapas para representação de dados estatísticos em <i>browsers</i>	30
2.5 Representação de dados estatísticos em mapas	32
2.5.1 Fenómenos Qualitativos.....	33
2.5.2 Fenómenos Quantitativos.....	34
2.5.3 Variação na intensidade da cor.....	35
2.7 Realidade aumentada	35
2.7.1 Realidade aumentada aplicada ao turismo	36
2.7.2 Testes a plataformas de realidade aumentada	36
2.8 Integração com redes sociais	40
2.9 Soluções para criação de API's em Django	
3. Análise dos Requisitos	45
3.1 Atores	45
3.2 Requisitos funcionais do sistema	45

3.2.1 Definição de <i>user stories</i>	45
3.2.2 Modelo de casos de utilização	48
3.3 Requisitos não funcionais do sistema.....	54
3.4 Diagramas de casos de uso	55
3.4.1 Diagrama de casos de uso para a aplicação móvel.....	56
3.4.2 Diagrama de casos de uso para a gestão de conteúdos/Visualização de estatísticas	57
3.5 Conclusão.....	58
4. Definição da Arquitetura.....	59
4.1 Vista de alto nível da arquitetura	59
4.2 Requisito para a arquitetura	59
4.2.1 Sistema operativo	59
4.2.2 Funcionamento offline	60
4.2.3 Manutenção	61
4.2.4 Componente servidor	61
4.2.5 Componente Mobile	62
4.2.6 Componente de visualização de estatísticas.....	62
4.2.7 Sincronização dos dados	63
4.3 Vistas da arquitetura.....	66
4.3.1 Vista lógica	67
4.3.2 Vista de desenvolvimento.....	68
4.4 Modelo de dados	69
4.4.1 Diagrama de classes	70
4.4.2 Diagrama de entidades/relacionamentos	71
4.5 API REST	72
4.5.1 Métodos implementados de raiz:.....	73
4.5.2 Métodos adaptados da API existente.....	73
4.5.3 Autenticação.....	74
5. Planeamento e Metodologia.....	75
5.1 Calendário do plano de Trabalho definido inicialmente	75
5.1.1 Calendário do primeiro semestre	75
5.1.2 Calendário previsto inicialmente para o segundo semestre	77
6. Desenvolvimento.....	82

6.1 Desenvolvimento da componente servidor	82
6.2 Desenvolvimento da Aplicação móvel.....	83
6.2.1 Prototipagem da aplicação móvel	83
6.2.2 Desenvolvimento da aplicação Android	84
6.2.3 Principais problemas encontrados e soluções.....	92
6.4.1 Estrutura do código da aplicação.....	94
6.4.2 Capturas de ecrã da aplicação	95
6.5 Outras aplicações desenvolvidas no decorrer do estágio.....	96
7. Testes	98
7.1 Testes para o servidor	98
7.2 Testes para a aplicação móvel.....	102
7.2.1 Testes de usabilidade	102
7.2.2 Testes em ambiente real	104
7.2.3 Testes à precisão da realidade aumentada	108
7.2.4 Testes aos mecanismos de sincronização de dados	108
7.3 Testes para a aplicação de visualização de estatísticas.....	110
8. Conclusão e Trabalho Futuro.....	116
Anexos	125

Lista de Figuras

Figura 1: <i>Screenshots</i> da aplicação COMPASS [8].....	18
Figura 2: Interface da aplicação Gullivers Genie [39].....	19
Figura 3: Interface da aplicação Lol@ [36].....	20
Figura 4: Interface da aplicação CyberGuide [3].....	20
Figura 5: <i>Screenshots</i> da aplicação TripAdvisor[10].....	21
Figura 6: <i>Screenshots</i> da aplicação America's National Parks [61].....	21
Figura 7: <i>Screenshot</i> da aplicação WalkMe [62].....	22
Figura 8: <i>Screenshot</i> da aplicação MyOrpheo [52].....	22
Figura 9: <i>Screenshot</i> da aplicação Regensburg [43].....	23
Figura 10: <i>Screenshots</i> da aplicação Multimedia Museum Guide.....	23
Figura 11: <i>Screenshots</i> da aplicação Mesh-t.....	24
Figura 12: <i>Screenshots</i> da aplicação Mobile Travel Guide.....	24
Figura 13: Representação das componentes de um serviço baseado na localização (adaptado de [4]).....	27
Figura 14: Representação de um mapa recorrendo ao Google Maps (imagem da esquerda), OpenStreetMaps(figura do centro) e Bing Maps(figura da direita).....	30
Figura 15: Exemplos de mapas implementados pelo PolyMaps (figura da esquerda), OpenLayers (figura do meio) e GeoTools (figura da direita).....	31
Figura 16: Quadro de variáveis visuais [20].....	33
Figura 17: Exemplificação da representação de fenómenos qualitativos [20].....	34
Figura 18: Exemplificação da representação de fenómenos quantitativos [20].....	34
Figura 19: Exemplificação da utilização de cores na representação de dados em mapas.....	35
Figura 20: Representação de um teste efetuado com a plataforma Qualcomm.....	37
Figura 21: Representação de um teste efetuado pela plataforma Wikitude.....	38
Figura 22: Representação de um teste efetuado com a plataforma Mixare.....	39
Figura 23: Exemplo da utilização de Graph API.....	41
Figura 24: Diagrama de casos de uso da aplicação móvel.....	56
Figura 25: Diagrama de casos de uso para a parte de administração/gestão de conteúdos.....	57
Figura 26: Diagrama de alto nível da arquitetura.....	59

Figura 27: Camadas da Arquitetura Android	60
Figura 28: Diagrama de sequência para a sincronização dos dados.....	64
Figura 29: Diagrama de sequência para a sincronização das imagens.....	65
Figura 30: Vista lógica da arquitetura	68
Figura 31: Vista de desenvolvimento da arquitetura	69
Figura 32: Representação do diagrama de classes	70
Figura 33: Representação do modelo de dados	71
Figura 34: Comparação de um <i>mockup</i> criado pelo autor (figura da esquerda) com um <i>mockup</i> criado pelo <i>designer</i> (figura da direita).....	84
Figura 35: Estrutura do código de uma aplicação Android	85
Figura 36: Resultado final para a listagem das plantas(figura da esquerda) e a utilização do mapa na aplicação (figura da direita)	91
Figura 37: Representação de uma captura de ecrã para a utilização da realidade aumentada.....	92
Figura 38: Resultado para a representação de dados estatísticos na aplicação final.	96
Figura 39: formato de resposta do SUS.....	103
Figura 40: Representação de um código QR [44].....	126
Figura 41: Representação de um código Microsoft Tag [49].....	126
Figura 42: Representação de um código DataMatrix [49]	126
Figura 43: Representação de um código PDF417 [56].....	127
Figura 44: Representação de um código MaxiCode.....	127
Figura 45: Representação de um código ScanLife [7].....	127
Figura 46: Menu principal da aplicação	128
Figura 47: <i>Mockup</i> criado pelo autor para o Menu Principal.....	129
Figura 48: ecrãs da listagem das plantas(esquerda), filtros da listagem(centro) e BI da planta(direita)	129
Figura 49: Mockups criados pelo autor para a listagem das plantas e BI das mesmas	130
Figura 50: prototipagem do mapa da aplicação.....	130
Figura 51: <i>Mockup</i> criado pelo autor para a utilização do mapa.....	131
Figura 52: prototipagem da componente de rotas da aplicação	131
Figura 53: <i>Mockups</i> criados pelo autor para a componente de rotas	132

Figura 54: prototipagem do guia real(realidade aumentada) da aplicação	132
Figura 55: <i>Mockup</i> criado pelo autor para a componente de realidade aumentada	133
Figura 56: Representação do menu principal da aplicação (figura do meio), da listagem das plantas (figura da esquerda) e do menu de filtros à listagem das plantas (figura da direita)	133
Figura 57: Representação do BI de uma planta (figura da esquerda), escolha do modo de visualização (figura do meio) e escolha da rede social para partilhar (figura da direita)	134
Figura 58: Representação da listagem das rotas (figura da esquerda), da criação de uma nova rota (figura do meio) e da escolha da ordem dos pontos (figura da direita)	134
Figura 59: Representação da utilização do mapa 2D na aplicação.....	135
Figura 60: Resultados para a utilização da realidade aumentada numa rota	136
Figura 61: Resultados para a utilização da realidade aumentada (guia geral)	136
Figura 62: Resultados para a realidade aumentada na pesquisa de uma determinada planta.....	137
Figura 63: Resultados para a partilha de dados na rede social Twitter	137
Figura 64: Processo de partilha de informação da rede social Facebook.....	138
Figura 65: Resultado para a partilha de informação na rede social Facebook.....	138
Figura 66: Representação do vídeo introdutório (figura da esquerda) e menu principal (figura da direita).....	139
Figura 67: Representação da escolha da personagem a seguir	139
Figura 68: <i>Printscreen</i> do BI da personagem escolhida	140
Figura 69: <i>Printscreen</i> do mapa utilizado na aplicação'	140
Figura 70: <i>Printscreen</i> da galeria de conteúdos multimédia de um ponto de interesse	141
Figura 71: <i>Printscreen</i> do diário de bordo da aplicação.....	141
Figura 72: Ecrã de login da aplicação.....	142
Figura 73: Representação de uma das métricas da aplicação	142
Figura 74: Representação completa da escolha dos filtros	143
Figura 75: Representação dos resultados para a filtragem dos dados	144

Lista de Tabelas

Tabela 1: Quotas de Mercado das plataformas móveis no ano 2011 [50].....	17
Tabela 2: Quotas de Mercado das plataformas móveis no ano 2012 [50].....	17
Tabela 3: Previsão das Quotas de Mercado das plataformas móveis para o ano 2016 [51].....	17
Tabela 4: Revisão dos guias móveis aplicados ao turismo	26
Tabela 5: Revisão das bibliotecas de mapas para aplicações móveis.....	32
Tabela 6: Revisão de bibliotecas de mapas para aplicações <i>browser</i>	32
Tabela 7: Revisão das plataformas de realidade aumentada.....	39
Tabela 8: Comparação de características para as ferramentas analisadas	41
Tabela 9: Comparação de características para as ferramentas de criação de APIs REST em Django	43
Tabela 10: Comparação de características para os motores de bases de dados [32]	44
Tabela 11: Componentes do <i>template</i> da definição dos requisitos.....	48
Tabela 12: Categorias de priorização de requisitos	49
Tabela 13: Categorias de tipos de requisitos	49
Tabela 14: Resumo dos métodos da API.....	83
Tabela 15: Principais ficheiros da package com.example.ar.....	85
Tabela 16: Principais ficheiros da package com.example.botanico	86
Tabela 17: Principais ficheiros da package com.example.classes.....	87
Tabela 18: Principais ficheiros da package com.example.db.....	87
Tabela 19: principais ficheiros da package com.example.parsejson	88
Tabela 20: Principais ficheiros da package com.example.rest.....	88
Tabela 21: Principais ficheiros da package com.example.twitter.....	88
Tabela 22: Resultados para a iteração de testes para o servidor.....	102
Tabela 23: Resultados para a primeira iteração de testes em ambiente real para a aplicação móvel	106
Tabela 24: Resultados para a segunda iteração de testes em ambiente real para a aplicação móvel	108
Tabela 25: Resultados para a iteração de testes para a sincronização dos dados	110

Tabela 26: Resultados obtidos para a primeira iteração de testes da componente de visualização de estatísticas	114
Tabela 27: resultados obtidos para a segunda iteração de testes da componente de visualização de estatísticas	115
Tabela 28: Revisão das tecnologias existentes para códigos bidimensionais.....	128

Termos e Acrónimos

AR	Augmented Reality
API	Application Programming Interface
CSS	Cascading Style Sheets
GIS	Geographic Information System
GPRS	General Packet Radio Service
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
MTV	Model – Template - View
MVC	Model – View - Controller
ORM	Object-Relational Mapping
OWL	Web Ontology Language
PDA	Personal Digital Assistant
POI	Point Of Interest
REST	Representational State Transfer
SBL	Serviço Baseado na Localização
SDK	Software Development Kit
SQL	Structured Query Language
SVG	Scalable Vector Graphics
UMTS	Universal Mobile Telecommunication System
WLAN	Wireless Local Area Network
WWW	World Wide Web
XML	Extensible Markup Language
YAML	Yet Another Markup Language

Capítulo 1

Introdução

Este capítulo inicial serve de introdução ao estágio, descrevendo o enquadramento do estágio assim como os objetivos do mesmo e a sua motivação.

1.1 Enquadramento

A proliferação e a crescente utilização dos *smartphones* vieram revolucionar a maneira como se vive atualmente. Tem-se verificado um grande desenvolvimento desta área tecnológica, que está cada vez mais acessível ao cidadão comum. Consequentemente, isto tem levado a um aumento da utilização de aplicações móveis nas mais diversas situações e áreas.

Estas aplicações tiram vantagem das capacidades dos dispositivos móveis (acelerómetro, GPS, ligação 3G, entre outras) para fornecer informação útil e contextualizada ao utilizador, de forma ubíqua.

Portugal é um importante destino turístico devido aos maravilhosos recursos ambientais que dispõe (praias, rios, serras, entre outros). Estes recursos proporcionam uma beleza e riqueza ambiental única. Nos últimos tempos tem-se verificado o aparecimento de iniciativas que promovem esta riqueza (7 Maravilhas Naturais de Portugal ®, 7 Maravilhas Praias de Portugal ®)

Apesar disto uma grande parte da população ainda desconhece a verdadeira riqueza ambiental do nosso país.

Este estágio insere-se no âmbito do projeto “*Pervasive Tourism/Smart Tourism (TOUREMOTION WALK Natureza & Sol e Mar)*” da empresa Ubiwhere,¹ na área do Turismo. A Ubiwhere providencia serviços de consultoria e desenvolvimento de software para Redes Heterogéneas e Redes de Próxima Geração. As suas atividades envolvem ainda a investigação na área da Computação Ubíqua e a criação de serviços inovadores para dispositivos móveis. Especial ênfase é dada à área do Turismo, através de aplicações baseadas na localização dos utilizadores e que lhes enriqueça a experiência. O projeto TOUREMOTION WALK consiste num guia outdoor para *smartphone*. O guia fornece a um utilizador experiência multimédia sobre pontos de interesse, serviços e eventos no decurso de uma visita a uma cidade ou região. O projeto desenvolvido neste estágio consiste num guia turístico móvel para *smartphone*, para o Jardim Botânico de Coimbra², de modo a tornar a visita o mais agradável possível ao visitante. Assim, a aplicação a desenvolvida, pela sua natureza, enquadra-se no projeto TOUREMOTION WALK.

¹ www.ubiwhere.com

² www.uc.pt/jardimbotanico

³ <http://developer.android.com/index.html>

1.2 Objetivos

O objetivo deste estágio consiste no desenvolvimento de um guia turístico móvel para o Jardim Botânico de Coimbra para a plataforma móvel Android. A aplicação deve tirar partido das potencialidades dos dispositivos móveis (acelerómetro, GPS, ligação à internet, entre outros) de modo a apresentar aos visitantes uma solução que torne a visita no interior do Jardim o mais agradável possível. Apresentam-se os principais objetivos do trabalho desenvolvido:

- Estudo de soluções existentes vocacionadas para diversas plataformas móveis e que estejam relacionadas com a natureza e turismo.
- Estudo de soluções tecnológicas que podem ser incorporadas na aplicação a desenvolver. Enquadram-se aqui a realidade aumentada, a utilização de mapas, entre outros;
- Adaptação da solução *Pervasive Tourism* da empresa para a aplicação do Jardim Botânico;
- Implementação do guia turístico para o Jardim Botânico utilizando as soluções tecnológicas analisadas. Com isto, pretende-se oferecer ao visitante do Jardim, utilizador da aplicação móvel, um experiência mais agradável durante a sua visita.
- Implementação de uma componente de visualização de estatísticas, recorrendo a uma página web, utilizando dados recolhidos pela aplicação móvel. Com isto, torna-se possível visualizar quais as plantas que são mais/menos visitadas pelos utilizadores.

1.3 Motivação

A principal motivação deste estágio é tornar a visita do Jardim muito mais agradável para o visitante, recorrendo à aplicação móvel. Outro elemento motivador é o facto de a aplicação oferecer ao utilizador a possibilidade de visualizar elementos e obter informação na aplicação através da realidade aumentada. Torna-se uma alternativa à apresentação da informação em duas dimensões e para além disso, torna a aplicação mais completa e agradável.

Para o estagiário, este estágio é particularmente motivante uma vez que a área da programação para *smartphones*, recorrendo à mobilidade é uma área que o agrada muito, sendo a área que mais interesse despertou ao longo do percurso académico. Outro fator importante de motivação é o facto de o estágio realizar-se numa empresa, oferecendo um primeiro contacto com o mundo do trabalho.

1.4 Estrutura do documento

Este documento está estruturado da seguinte forma: no capítulo 1, é feita uma breve introdução, definindo o enquadramento do estágio, os objetivos do mesmo assim como a motivação que está por detrás deste estágio. O capítulo 2 é dedicado ao estudo do estado da arte feito pelo autor, onde foram analisadas as tecnologias utilizadas no projeto e retiradas conclusões das mesmas. O capítulo 3 é dedicado à análise e levantamento de requisitos da aplicação. O capítulo 4 é dedicado à definição da arquitetura do sistema implementado. No capítulo 5 é apresentada a calendarização do estágio, identificando as atividades desenvolvidas assim como as entregas planeadas. Também é apresentada a metodologia de desenvolvimento que foi seguida assim como alguns artefactos que a complementaram. O capítulo 6 é dedicado ao desenvolvimento das componentes criadas. O capítulo 7 apresenta os testes efetuados para validar as componentes desenvolvidas. Finalmente, o capítulo 8 apresenta as conclusões retiradas pelo autor assim como o trabalho futuro identificado pelo mesmo.

Capítulo 2

Estado da Arte

Este capítulo aborda o Estudo do Estado da Arte. O capítulo está estruturado da seguinte maneira: A secção 2.1 descreve o estado atual das quotas de mercado das principais plataformas móveis, assim como previsões para o ano 2015; A secção 2.2 destina-se à revisão literária sobre aplicações que estão ou já estiveram no mercado, relacionadas com o turismo. A secção 2.3 apresenta as características essenciais para a criação de sistemas baseados na localização, sendo este o tipo de sistema implementado no decorrer do estágio; Na secção 2.4 são apresentadas as principais soluções de mapas utilizadas em aplicações móveis. A secção 2.5 aborda o tema da representação de dados estatísticos em mapas, permitindo demonstrar quais as melhores maneiras para representar dados quantitativos e qualitativos em mapas, recorrendo à cartografia; São apresentadas, na secção 2.6, bibliotecas para implementação de realidade aumentada nas aplicações móveis, algumas delas testadas pelo autor. A secção 2.7 dedica-se ao estudo de ferramentas para exploração de dados presentes nas redes sociais. Na secção 2.8 encontra-se um estudo comparativo sobre as principais bibliotecas para a criação de APIs REST utilizando a *framework* Django. Finalmente, a secção 2.9 apresenta um estudo comparativo de motores de bases de dados.

2.1 Plataformas Móveis

Nesta secção são analisadas as principais plataformas móveis presentes no mercado. Muitas vezes, um dos objetivos quando se cria uma aplicação móvel, é que esta alcance o maior número possível de pessoas. Por isso, é feita aqui uma análise das quotas de mercado das plataformas móveis presentes no mercado, com vista a determinar qual a plataforma mais utilizada. Como é possível ver nos dados presentes na Tabela 1 [50], o sistema operativo Android é aquele que foi mais utilizado em 2011 deixando muito para trás os restantes sistemas operativos. No segundo quarto de ano de 2012, a plataforma Android continuou bem a frente das restantes plataformas, como é possível ver na Tabela 2 [50]. Existe uma previsão (feita no mês de Junho do ano 2012) que aponta para a continuação da dominação do sistema operativo Android em 2016, como se pode ver na Tabela 3. [51].

Plataforma	Share(%)
Android	49,2 %
iOS	18,8 %
Symbian	16,5 %
BlackBerry	10,3 %
Windows Phone	1,8 %
Others	3,3 %

Tabela 1: Quotas de Mercado das plataformas móveis no ano 2011 [50]

Plataforma	Share(%)
Android	68,8 %
iOS	18,8 %
Symbian	3,3 %
BlackBerry	4,5 %
Windows Phone	2,5 %
Others	2,1 %

Tabela 2: Quotas de Mercado das plataformas móveis no ano 2012 [50]

Plataforma	Share(%)
Android	52,9 %
iOS	19,0 %
BlackBerry	5,9 %
Windows Phone	19,2 %
Others	3,0 %

Tabela 3: Previsão das Quotas de Mercado das plataformas móveis para o ano 2016 [51]

Após análise das plataformas utilizadas no mercado, observa-se que a escolha da plataforma Android para o desenvolvimento da aplicação é uma escolha acertada. O sistema operativo Android dominou as vendas no ano de 2011 prevendo-se a continuação dessa predominância em 2016. Portanto, a escolha do desenvolvimento da aplicação para a plataforma Android parece acertada, uma vez que, de acordo com as tabelas 1, 2 e 3, a aplicação poderá chegar a uma maior quantidade de utilizadores. Outro fator importante,

da parte do programador, encontra-se na comunidade. A comunidade Android³ oferece muito apoio aos programadores com recurso a tutoriais e documentação e fóruns de discussão e dúvidas.

2.2 Guias turísticos móveis baseados na localização

Apresentam-se nesta secção algumas soluções para guias turísticos móveis existentes no mercado. Foram analisadas algumas soluções consideradas como sendo populares na literatura, bem como algumas soluções que o autor considerou relevante para o estudo, tendo em conta o objetivo final do estágio. Para além de guias turísticos móveis, foram também analisados guias turísticos direcionados para determinados pontos de natureza, como por exemplo parques naturais ou museus.

2.2.1 Soluções Analisadas

COMPASS

Diminutivo de *C*ontext-aware *M*obile *P*ersonal *A*SSistant, esta aplicação permite aos turistas obterem serviços e recomendações baseadas no contexto em cada momento. Para a realização da personalização de conteúdos, faz o uso de um registo que contém informação acerca de *third party services*, fornecendo informação sobre restaurantes e museus. Para a descrição de serviços, é usada tecnologia de Web Semântica, como por exemplo *Web Ontology Language (OWL)*. O sistema adquire e fornece informação ao utilizador acerca do seu contexto, contactando o serviço de contexto apropriado [1] [8]. A informação de localização é obtida com recurso às funcionalidades do GPS. O utilizador beneficia de um sistema de recomendações que segue o conceito de adaptação, ou seja, à medida que o utilizador vai traçando o seu perfil, as recomendações vão adaptar-se ao perfil do utilizador. Para além disso, a aplicação faz o uso de vários serviços externos de mapas através de interfaces proprietárias, tal como *Microsoft Mappoint* para mapas regulares (um serviço de mapas que fornece fotografias e um serviço de mapas que fornece mapas cadastrais). A plataforma faz o uso de conexões a redes 3G, requerendo uma conexão permanente à rede internet.



Figura 1: Screenshots da aplicação COMPASS [8]

³ <http://developer.android.com/index.html>

CRUMPET

CRUMPET é um projeto europeu que tem como objetivo a “*Creation of User-friendly Mobile Services Personalized for Tourism*”, através da utilização de agentes. O utilizador pode requisitar informações e recomendações sobre atrações para turistas, restaurantes e passeios. O sistema é capaz de fornecer dicas pró-ativas quando o utilizador se encontra perto de um sítio que é do seu interesse, suportando mapas interativos, mostrando a posição do utilizador bem como locais interessantes para ele. *CRUMPET* considera a localização, o dispositivo, a rede, assim como as propriedades do contexto. Esta aplicação oferece serviços para *nomadic-users* [1].

GUIDE

Esta aplicação pertence à área de *location-based service*. A aplicação permite fornecer aos turistas informação baseada no contexto de uma cidade, utilizando um PDA. O sistema é baseado numa arquitetura cliente/servidor, utilizando um Fujitsu TeamPad 7600 como terminal. Esta aplicação fornece um modelo lógico de contexto mais detalhado, distinguido mesmo entre *personal context* em termos de informação recolhida acerca do utilizador (i.e. preferências, localização atual, histórico de atrações já visitadas) e entre *environmental context* que inclui a informação acerca de atrações (i.e. ligações entre as atrações, horários de abertura e fecho, etc...). No que toca a informação de posicionamento, esta aplicação usa a rede WLAN, à qual o utilizador está ligado para obter a sua localização. A aplicação fornece aos utilizadores informações sobre locais a visitar, bem como um mapa e a possibilidade de criar uma rota [1].

Gullivers Genie

Trata-se de uma aplicação baseada em agentes que criam as suas deduções num estado mental, seguindo o paradigma de Crenças, Desejos e Intenções. A aplicação permite efetuar a apresentação dos pontos de interesse aos utilizadores. Para além da apresentação de pontos de interesse, a aplicação fornece também informação associada aos *POI's*, como áudio, imagens ou *links*. Também permite ao utilizador adicionar comentários aos *POI's* visitados. A aplicação cliente corre num PDA equipado com GPS e suporte a comunicações por *wireless* [1] [39].



Figura 2: Interface da aplicação Gullivers Genie [39]

É o resultado de um protótipo para aplicação móvel baseada na localização por GPRS e UMTS, oferecendo aos turistas informação multimédia acerca da cidade de Viena. O seu principal foco encontra-se no suporte aos turistas durante as suas visitas oferecendo rotas predefinidas assim como informações acerca de POI's. A aplicação destina-se apenas a um telemóvel. A aplicação determina a posição do utilizador e fornece um *personal guided tour* ao mesmo. Além disto, fornece conteúdos específicos acerca da localização do utilizador. [1] [36].



Figura 3: Interface da aplicação Lol@ [36]

Cyberguide

Descrito como sendo um dos primeiros guias móvel. Funciona tanto *indoor* como *outdoor*. Este guia fornece simples esquemas de mapas a preto e branco assim como serviços de informação de localizações *indoor* e *outdoor* pré-definidas. Todos os mapas e informações são estáticos e guardados no dispositivo móvel. O posicionamento *indoor* é resolvido com recursos a ondas infravermelhas enquanto o posicionamento *outdoor* é resolvido utilizando o GPS [2] [3].

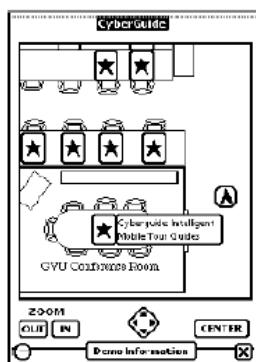


Figura 4: Interface da aplicação CyberGuide [3]

TripAdvisor

TripAdvisor é uma das aplicações mais populares no domínio dos guias turísticos. Através desta aplicação um utilizador pode pesquisar hotéis, restaurantes e atividades divertidas para fazer em locais próximos de onde se encontra. Permite sobretudo encontrar pontos turísticos em qualquer destino. É possível visualizar conteúdos multimédia atualizados (i.e. fotografias) uma vez que os utilizadores podem colocar fotografias, sendo portanto uma aplicação colaborativa do ponto de vista da introdução de informação. Baseia-se também em conteúdos inseridos pelas entidades gestoras da aplicação [10].

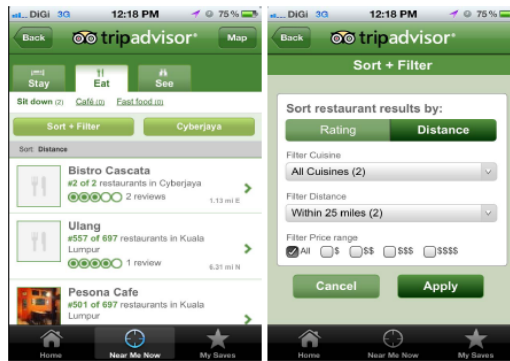


Figura 5: *Screenshots* da aplicação TripAdvisor[10]

America's National Parks

Esta aplicação fornece informação sobre cada um dos 58 parques nacionais que se encontram nos Estados Unidos. Esta aplicação indica ao utilizador como chegar a um determinado parque. Fornece um mapa de cada parque, um mapa topológico para caminhadas assim como informação sobre as atividades dos parques. É possível partilhar seus próprios passeios com outras pessoas ou então armazená-los localmente para uso futuro. Uma vez carregados, os passeios podem ser usados sem conectividade de rede. A aplicação está integrada com o *Flickr* e o *Google* [61].



Figura 6: *Screenshots* da aplicação America's National Parks [61]

WalkMe

Trata-se de uma aplicação móvel, para Android, que guia os turistas na descoberta das maravilhas naturais da Ilha da Madeira. Esta aplicação permite visualizar percursos no mapa para serem seguidos. A aplicação permite obter informação no mapa sobre os alojamentos e percursos pedestres próximos. A aplicação permite obter a localização do utilizador em tempo real com recurso ao GPS do dispositivo móvel. Existe integração da aplicação com os alojamentos da Madeira Rural, fornecendo informação útil como o número de quartos e camas, preços e localização. Existe a possibilidade de partilhar cada percurso nas redes sociais [62].



Figura 7: *Screenshot* da aplicação WalkMe [62]

MyOrpheu

Trata-se de uma aplicação móvel que fornece aos visitantes soluções emocionais assim como educacionais em equipamentos pessoais (*MP3 players, smartphones, tablets*), para museus, galerias de arte, passeios pela cidade, parques e zonas industriais. A aplicação existe para as plataformas Android (OrpheoDroid) e iOS (iOrpheo). Graças a esta aplicação é muito fácil implementar e gerir aplicações multimédia Android, tanto nos *smartphones* como em *tablets*, em *sites* culturais. Oferece uma arquitetura flexível que permite selecionar recursos específicos necessários ao utilizador. Uma funcionalidade importante da aplicação é a implementação de realidade aumentada. A aplicação é capaz de fornecer experiências 2D e 3D para os museus, exposições e locais culturais. De entre todas as aplicações desenvolvidas, destaca-se o *Gardens of International Perfume Museum* que consiste num guia áudio e multimédia sobre um museu de jardins, permitindo descobrir diferentes tipos de flores e perfumes. [52]



Figura 8: *Screenshot* da aplicação MyOrpheo [52]

Regensburg iPhone app:

Trata-se de uma aplicação móvel para iPhone disponibilizada pela oficina de turismo da cidade de Regensburg, em Alemanha. Utilizando esta aplicação, um visitante pode visualizar fotos sobre os sítios mais interessantes da cidade, efetuar uma reserva num hotel, visualizar o calendário de eventos da cidade, efetuar visitas guiadas e visualizar dados sobre os museus. Além disso, a aplicação fornece um sistema de mapas, permitindo ao utilizador encontrar o seu caminho para os pontos de interesse. A aplicação está

integrada com as redes sociais, permitindo partilhar fotografias ou frases no Facebook ou Twitter. Outra componente interessante da aplicação é a utilização de realidade aumentada, aplicada nos locais históricos da cidade [43].



Figura 9: Screenshot da aplicação Regensburg [43]

2.2.2 Soluções existentes da empresa Ubiwhere

Apresentam-se nesta secção algumas das soluções da empresa Ubiwhere.

Ubiwhere Multimedia Museum Guide

O primeiro guia turístico da *Ubiwhere* para museus. Esta aplicação serve principalmente de guia turístico para os visitantes de museus, com um modelo de implementação sensível ao contexto do utilizador. Tira partido de um conjunto de tecnologias de localização em espaços fechados, como RFID e Ultrassons. Esta aplicação é direcionada para o Museu da Chapelaria, em São João da Madeira. A aplicação possui módulos de posicionamento *Indoor* que permitem ao visitante conhecer a sua localização no interior do museu, conhecer a localização das peças em exposição, identificar a peça que está a ver.



Figura 10: Screenshots da aplicação Multimedia Museum Guide

Mesh-t

Esta solução consiste num conjunto de aplicações para o sector do turismo: Um guia turístico móvel (para dispositivos móveis) baseado no contexto (principalmente na

localização e preferências de cada turista), uma solução turística enriquecida baseada em superfícies interativas; Um guia multimédia para museu com reconhecimento automático de itens em exposição. A aplicação móvel permite obter a localização no interior do Museu bem como a localização de itens no interior do Museu. Permite determinar a peça que o utilizador está a ver através do reconhecimento de marcadores (QR Codes). Oferece informação multimédia de vários tipos (texto e vídeos) sobre os itens em exposição. Também é possível definir e seguir rotas para as visitas no interior do Museu.



Figura 11: Screenshots da aplicação Mesh-t

Ubiwhere Mobile Travel Guide

Trata-se de um guia turístico para posicionamento *outdoor* da Ubiwhere. Utiliza georreferenciação por GPS. Possui funcionalidades como definir POI's ou eventos e alertar o utilizador quando se encontra na sua vizinhança. Também permitem que o utilizador defina um perfil que, juntamente com as ações que toma, ajuda a afinar os alertas que lhe são mostrados, bem como sugerir localizações, serviços ou eventos que lhe possam interessar, de acordo com o perfil definido.



Figura 12: Screenshots da aplicação Mobile Travel Guide

2.2.3 Revisão de guias existentes

A tabela 4 classifica os guias analisados de acordo com um conjunto de características distintivas: exibição de características *context-aware*, a existência de um modo *offline* e a existência de realidade aumentada.

Aplicação	Origem	Principal foco	Real. Aum.	Modo off.
COMPASS	Context-aware recommendation system	Integração de <i>contexto-aware</i> e sistemas de recomendação em aplicações móveis para turistas	Não	Não
CRUMPET	<i>Mobile services</i> personalizados para o turismo	Suporte de utilizadores nómades através da disponibilização de serviços	Não	Não
GUIDE	Location-based services	Geração de informação para o turista através de <i>contexto-aware</i>	Não	Sim
Gullivers Genie	Inteligência artificial e sistemas de agentes	Disponibilizar conteúdo inteligente, baseado em agentes	Não	Sim
Lol@	Aplicação desenvolvida para dispositivos restritos, isto é aparelhos com GSM/UMTS	Aplicação UMTS	Não	Não
CyberGuide	Context-Aware	Disponibilização de conteúdos indoor através de infravermelhos e <i>outdoor</i> através do GPS.	Não	Não
TripAdvisor	Location-aware	Disponibilização de informação, oficial ou inserida pelos utilizadores, através de <i>Location-Aware</i> .	Não	Não
WalkMe	Location-Aware	Disponibilização de conteúdo através da localização a cada momento	Não	Sim
America's National Parks	Location aware	Disponibilização de conteúdos multimédia ao utilizador consoante a sua localização.	Não	Sim
MyOrpheo	Location-Aware	Aplicação móvel que fornece várias soluções aos utilizadores, baseada na localização.	Sim	Sim
Regensburg	Location-Aware	Aplicação móvel para iPhone que fornece e partilha conteúdos multimédia sobre a cidade, ao utilizador.	Sim	Não
Mesh-t	Location-aware	Conjunto de aplicações para o setor do turismo, fornecendo informação ao utilizador de acordo com o contexto e seu perfil.	Não	Não
Ubiwhere Multimedia Museum Guide	Location-Aware	Aplicação <i>indoor</i> utilizada em museus, sensível ao contexto do utilizador.	Não	Sim
		Guia turístico <i>outdoor</i> ,		

Ubiwhere mobile Travel Guide	Context-Aware	sensível ao contexto do utilizador assim como ao seu perfil, para efetuar recomendações de POI's.	Não	Não
------------------------------------	---------------	---	-----	-----

Tabela 4: Revisão dos guias móveis aplicados ao turismo

Comparação com a aplicação desenvolvida

Após efetuada uma análise a várias aplicações baseadas na localização, dirigidas para o turismo e natureza, verifica-se que todas elas têm componentes que estão integradas da aplicação móvel desenvolvida no decorrer do estágio. No entanto, não é possível encontrar uma aplicação que funciona em modo *offline*, que implemente realidade aumentada e integração com as redes sociais, ao mesmo tempo. Verifica-se então que a aplicação desenvolvida resulta num produto bastante completo.

2.3 Caracterização dos serviços baseados na localização

O termo “Serviços Baseados na Localização” (SBL) é um conceito que inclui aplicações que fazem uso da localização geográfica, utilizando coordenadas espaciais. Inclui-se nessa categoria qualquer solução onde o fator localização agrega valor a outros serviços disponibilizados em dispositivos móveis, através de uma rede. Este tipo de serviços está cada vez mais presente no quotidiano das pessoas, sendo utilizados em computadores, PDA's ou *smartphones*. Podemos classificar os serviços baseados na localização em duas categorias [38]:

- *Push services*: implicam que o utilizador recebe a informação como o resultado do seu paradeiro, sem ter que solicitá-lo ativamente. A informação pode ser enviada para o utilizador com ou sem o seu consentimento prévio.
- *Pull services*: em contraste com os *push services*, significa que o utilizador usa ativamente uma aplicação e, no seu contexto, “puxa” a informação a partir da rede. Essa informação pode ser sobre o descobrimento de uma localização (e.g., onde encontrar o cinema mais perto).

Atualmente, muitas das aplicações baseadas na localização fazem o uso de tecnologias GPS. A informação de localização de um utilizador de telemóvel pode ser utilizada por várias aplicações, com diferentes propósitos, para a tradicional navegação e rastreamento, navegação nas redes sociais em dispositivos móveis, pesquisas, recomendações, etc. As aplicações de navegação atuais fornecem mais funcionalidades do que as aplicações tradicionais. Em adição à navegação de um sítio A para um sítio B, estas aplicações mostram lugares específicos, como gasolinhas, restaurantes ou bancos. *Mobile social networking* aplicações permitem determinar se um amigo encontra-se nas proximidades para, por exemplo, ir ter com ele. *Mobile search systems* podem efetuar certas pesquisas tendo em conta a localização do utilizador, fornecendo informação mais específica e útil [4].

2.3.1: componentes principais:

Os serviços baseados na localização funcionam recorrendo a cinco componentes [4] [5]:

- **Dispositivos móveis:** representam os dispositivos utilizados para pedir e representar a informação. Os resultados podem ser apresentados sobre a forma de texto, imagem, áudio e vídeo. Relativamente ao tipo de equipamento, podem ser usados PDA's, *smartphones*, *tablets*, computadores portáteis, etc.
- **Redes de comunicação móvel:** trata-se da rede responsável por encaminhar os pedidos dos dispositivos móveis para o fornecedor de serviços.
- **Componente de posicionamento:** esta componente é responsável pela indicação do posicionamento do utilizador, podendo este ser disponibilizado através do *Global Positioning System (GPS)*, pelas redes sem fios ou antenas de rádio. Geralmente, os SBL fazem o uso de GPS.
- **Fornecedores de serviços e aplicações:** oferecem diversos serviços e são responsáveis pelo processamento dos pedidos que são feitos pelos assinantes. Os serviços oferecidos podem ser o cálculo do posicionamento, encontrar um percurso, procurar informação específica sobre locais, etc. Estes dados são processados pelos servidores de serviços, voltando depois para os utilizadores.
- **Fornecedores de dados e conteúdos:** normalmente, os fornecedores de serviços não guardam toda a informação pedida pelos utilizadores. Os fornecedores de conteúdos são responsáveis por recolher e guardar os dados geográficos, as informações sobre a localização, entre outros dados. Esses dados são tratados pelo servidor de serviços e mandados de volta para o utilizador.

A Figura 13 [4] mostra a interação entre as várias componentes identificadas acima, assim como o processo simplificado de um serviço baseado na localização. Em primeiro, o utilizador envia um pedido de serviço usando a aplicação que está correndo no dispositivo móvel (Passo 1). O pedido de serviço, com a localização do utilizador obtida pela componente de posicionamento (GPS neste caso), é enviado para o servidor de serviços recorrendo à rede de comunicação móvel (Passo 2). O servidor de serviços efetua pedidos à base de dados geográfica e outras bases de dados de maneira a obter a informação pretendida (Passo 3 e 4). Finalmente, a informação pedida é enviada de volta para o dispositivo móvel do utilizador, passando pela rede móvel de comunicação (Passo 5 e 6).

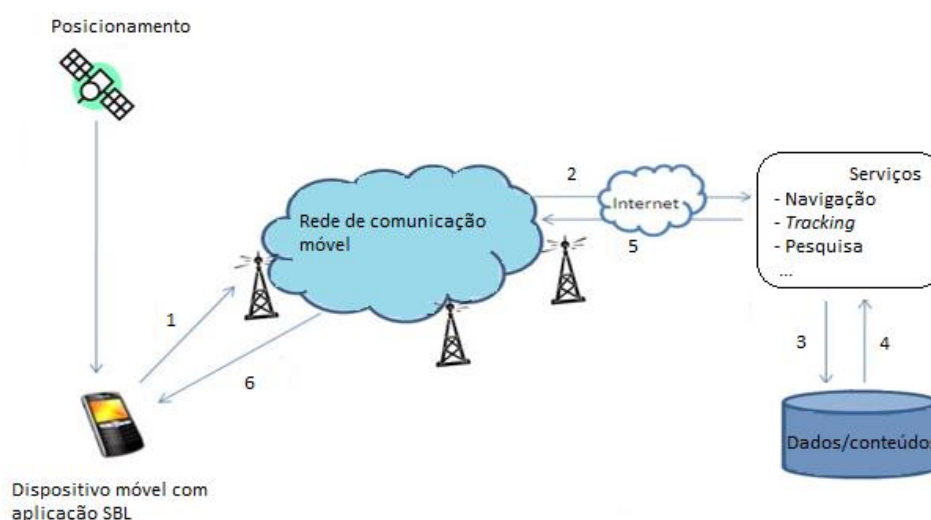


Figura 13: Representação das componentes de um serviço baseado na localização (adaptado de [4])

2.3.2 Fatores chave para o sucesso dos SBL

Analisa-se aqui os fatores chave para o sucesso de um serviço baseado na localização. Destacam-se os seguintes fatores [5]:

- **Confiança do cliente:** tornou-se crucial para o crescimento e sucesso do *mobile commerce*. A confiança é a componente chave para desenvolver relações adequadas com o cliente ou utilizador.
- **Experiência do utilizador:** Muitas companhias desenvolvem aplicações semelhantes baseadas na localização. No entanto, um dos fatores de sucesso encontra-se na experiência do utilizador com a utilização da aplicação. A aplicação tem que atender às necessidades dos utilizadores e, de preferência, melhorar a sua experiência tornando seu uso o mais agradável possível.
- **Custos dos serviços de dados sem fios:** Ter serviços de dados sem fios é crucial para os utilizadores acederem a SBL.
- **Qualidade da informação:** Muitos investigadores sugerem que a qualidade da informação é um fator de sucesso crítico para os serviços baseados na localização. Um bom modelo é um modelo de qualidade de informação, segundo o qual tem quatro dimensões principais: qualidade de conexão, qualidade dos conteúdos, qualidade de interação e qualidade contextual. A qualidade de conexão é atingida quando o utilizador consegue aceder o serviço móvel de internet sem ter interrupções na conexão e recebendo respostas muito rapidamente. A qualidade de conteúdo é obtida quando o conteúdo é guardado com objetivos e argumentos credíveis e quando a informação é útil e adequada para as tarefas do utilizador. A qualidade de interação é atingida quando os serviços móveis de internet fornecem métodos simples e eficientes, de interação, em termos de qualidade de usabilidade dos serviços móveis. A qualidade contextual é atingida quando o utilizador está em condições de usar a informação a qualquer momento e em qualquer lugar, sem ter que realizar grandes esforços para aceder à informação. A informação é fornecida no tempo e sítio certos.
- **Preço dos LBS e seu valor percebido:** a adoção dos SBL é afetada pela estratégia de preços aplicada a este tipo de serviços. A determinação do preço depende de vários parâmetros: fabricantes de telefones móveis, operadoras móveis, vendedores de aplicações, etc. Os preços que os utilizadores estão dispostos a pagar dependem do valor percebido, derivado dos serviços baseados na localização. A confiança é uma componente essencial para desenvolver um relacionamento rentável com o cliente.
- **Localização:** a localização reflete a capacidade técnica em determinar a atual localização física de um dispositivo móvel. A tecnologia de posicionamento usada em LBS para determinar a localização de utilizadores de dispositivos móveis em ambientes *outdoor* pode ser dividida em três categorias: *handset-based*, *cellular network-based* e *hybrid approaches*.
- **Personalização:** Muitos investigadores consideram que a personalização pode ser considerada como um fator chave para a adoção de serviços baseados na localização. Personalização é definida de maneira diferente consoante o contexto

em que se está inserido, podendo ser económico, marketing, informação de tecnologia, operações de procura.

2.3.3 Principais preocupações na implementação de SBL

O desenvolvimento de aplicações baseadas na localização requer alguns cuidados, nomeadamente nos seguintes tópicos [12]:

- **Privacidade:** Todo o ser humano tem direito à privacidade ou ser livre de intrusão ou distúrbios na vida pessoal. No caso de SBL, ou outras tecnologias de telecomunicações que lidam com a transformação de vários tipos de dados, torna-se essencial fornecer uma segurança adequada a esse tipo de dados para não serem mal utilizados por qualquer pessoa não-autorizada. Rastreamento e monitoramento de alguém sem o seu consentimento é puramente antiético, necessitando por isso de um alto nível de segurança. Os SBL podem ser muito benéficos mas também podem prejudicar a privacidade dos utilizadores.
- **Segurança:** De modo a manter a privacidade, o sistema de segurança deve ser forte. Toda a tecnologia tem tanto impacto positivo como impacto negativo na vida humana e os SBL também têm as suas falhas, localizando dados e permitindo facilmente acesso a pessoas que não tem autorização.
- **Tecnologias de posicionamento:** Nem todos os dispositivos móveis apresentam as mesmas tecnologias e, conseqüentemente, qualidade das mesmas, para a obtenção da localização das pessoas. Muitas vezes os dispositivos estão sujeitos a ruído, apresentando erros na obtenção da localização. É portanto fundamental minimizar os erros na obtenção da localização.

2.4 Principais soluções para utilização de mapas

Aplicações baseadas na localização, sejam elas móveis, web ou *desktop*, recorrem necessariamente a um mapa para a representação da localização do utilizador ou para representar a localização de quaisquer outros elementos [13]. Nesta secção são analisadas as principais API's existentes para a utilização de mapas nas aplicações móveis para *smartphone*. Foram analisadas as 3 soluções mais populares presentes no mercado que são o Bing Maps, Google Maps API e o Open Street Maps. Um dos aspetos mais importantes analisados aqui é a capacidade de cada uma destas API's em guardar mapas em modo *offline*, desligado de qualquer rede internet. Segue-se um estudo comparativo sobre cada uma das três soluções em causa.

Google Maps API: é uma das mais populares bibliotecas *Javascript* para a web, usada por mais de 350000 *websites* [13]. Esta biblioteca oferece recursos avançados: um poderoso sistema de roteamento (inclusive para andar de bicicleta, a pé ou de carro), *Street View*, construções em 3D, previsão do tempo e informações sobre o trânsito. Um grande problema com esta biblioteca da Google é que o utilizador tem muito pouco, ou nenhum controlo. Google Maps vem com três mapas de base: rua, satélite e terreno. Esta biblioteca permite guardar um mapa no *smartphone* mas de maneira limitada. É permitida guardar uma área de cerca de 26 quilómetros quadrados [47], predefinida pelo utilizador, ou seja, o utilizador tem que escolher manualmente a área do mapa que deseja guardar [71]. Com

isto, o utilizador consegue visualizar a área do mapa guardada sem recorrer a uma ligação á internet [46]. A área do mapa fica guardada na cache do dispositivo móvel. A Figura 14 (figura da esquerda) representa um mapa recorrendo à biblioteca Google Maps.

Bing Maps: Trata-se de uma biblioteca de mapas desenvolvida pela Microsoft. É vista como uma alternativa aos mapas do Google, especialmente para fornecer informações locais. Um recurso “exclusivo” do Bing Maps é a vista de “olho de pássaro”, que dá vistas aéreas de vários ângulos de perspetiva [13]. É possível utilizar esta API através de uma *API Rest Service*, bem como através de um SDK para as plataformas Android, iOS e Windows Phone. A principal desvantagem desta API é que pode tornar-se demasiado lenta no carregamento de imagens. A Figura 14 (figura da direita) representa um mapa recorrendo à biblioteca Bing.

OpenStreetMaps: Trata-se de um projeto que pretende construir um mapa livre de todo o mundo. Tal como a Wikipédia, o OpenStreetMap beneficia da entrada de dados de milhares de pessoas ao redor do mundo. A comunidade recolhe dados sobre estradas, ferrovias, ruas, rios e até mesmo ciclovias. Na realidade, o OpenStreetMap não é um mapa, mas uma base de dados. Nela, são gravados todos os objetos individuais que irão aparecer no mapa após passarem por um processo chamada renderização. [14]. Ao beneficiar da entrada de dados de milhares de pessoas ao redor do mundo, podem acontecer situações de inserção de dados “falsos”, constituindo uma desvantagem desta ferramenta. A Figura 14 (figura do meio) representa um mapa recorrendo ao OpenStreetMaps.

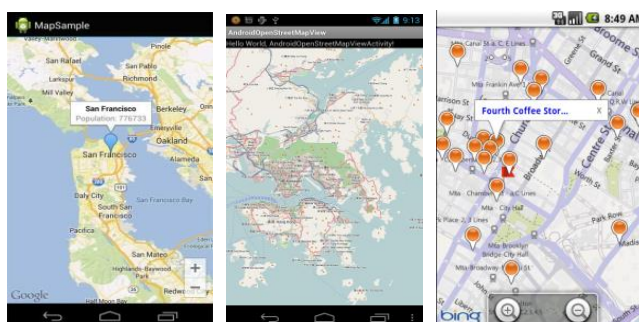


Figura 14: Representação de um mapa recorrendo ao Google Maps (imagem da esquerda), OpenStreetMaps(figura do centro) e Bing Maps(figura da direita)

Soluções de mapas para representação de dados estatísticos em *browsers*

As soluções analisadas acima foram estudadas com o objetivo de determinar a melhor biblioteca para a utilização de mapas na aplicação móvel a desenvolver. Nesta secção, foram analisadas API's para representação de mapas, mas desta vez tendo como objetivo determinar a melhor biblioteca capaz de representar dados estatísticos em mapas, utilizando um *browser* para a representação da informação. Esta análise foi feita tendo como objetivo determinar a melhor solução para a criação de uma componente de visualização de dados estatísticos num mapa, suportando a visualização de dados recolhidos pela a aplicação móvel a desenvolver durante o estágio. Foram analisadas as três seguintes API's:

Geo Tools: Trata-se de uma biblioteca *open-source* java para criação de mapas. Esta ferramenta disponibiliza métodos para a manipulação de dados geoespaciais como, por exemplo, para implementar Sistemas de Informação Geográfica (GIS) [15]. As principais funcionalidades da biblioteca são as seguintes: Definição de interfaces para os principais conceitos e estruturas de dados espaciais; Acesso a dados de GIS em arquivos de vários formatos e bases de dados espaciais. A Figura 15 (figura da direita) representa um mapa recorrendo à ferramenta Geo Tools.

OpenLayers: Trata-se de uma biblioteca Javascript para representação de mapas em aplicações web. Esta biblioteca torna fácil a ação de colocar mapas dinâmicos em páginas web. Permite a representação de vários tipos de marcadores em mapas. OpenLayers foi desenvolvido com o intuito de usar informação geográfica de vários tipos. A ferramenta é livre de custos para o programador [16]. A Figura 15 (figura do meio) representa um mapa recorrendo à ferramenta OpenLayers.

PolyMaps: Ao contrário de outras API's de mapa, Polymaps usa *Scalable Vector Graphics* (SVG) para exibir informação geográfica, renderizando diretamente os dados no navegador. Dados de mapas podem ser carregados usando formatos padrão como GeoJSON⁴ e renderizados diretamente. A utilização de SVG permite usar regras CSS para moldar componentes de mapas como estradas e fronteiras, incluindo mudando o estilo “*on the fly*”. Além disso, permite o uso de transições e animações SVG nos mapas [17]. A Figura 15 (figura da esquerda) mostra exemplos de mapas implementados pelo PolyMaps.

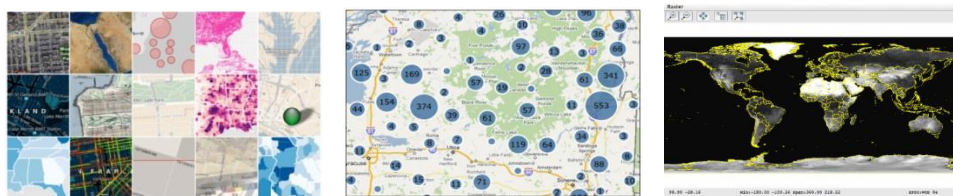


Figura 15: Exemplos de mapas implementados pelo PolyMaps (figura da esquerda), OpenLayers (figura do meio) e GeoTools (figura da direita)

Polymaps torna fácil a visualização de grandes conjuntos de dados de informação geográfica (por exemplo, a densidade populacional, as estatísticas de criminalidade ou padrões de voto). O nível de detalhe é controlado ajustando os dados geográficos, em vez de ajustar imagens renderizadas, para que possa ser alterada dinamicamente.

A principal desvantagem do uso de SVG é que ele não é totalmente suportado em todos os navegadores móveis e, se suportado, usa um enorme poder de processamento [45].

São apresentadas duas tabelas resumindo algumas características das ferramentas analisadas. A Tabela 5 refere-se às soluções analisadas para a representação de mapas em aplicações móveis enquanto que a Tabela 6 refere-se as soluções analisadas para a representação de dados estatísticos em mapas, num browser.

⁴ <http://www.geojson.org/>

Aplicação	Gratuito	Funcionamento offline	Street View
Google Maps API	Sim	Sim (Limitado)	Sim
Bing Maps	Sim	Não	Sim
OpenStreet Maps	sim	Sim	Não

Tabela 5: Revisão das bibliotecas de mapas para aplicações móveis

Aplicação	Gratuito	Funcionamento browser	Utilização de SVG
GeoTools	Sim	Sim	Não
PolyMaps	Sim	Sim	Sim
OpenLayer	Sim	Sim	Sim

Tabela 6: Revisão de bibliotecas de mapas para aplicações *browser*

Após a análise efetuada sobre as soluções mais populares oferecidas para implementação de mapas em aplicações móveis, é possível concluir que o OpenStreetMaps é a solução mais adequada para a representação de mapas, para utilizar na aplicação desenvolvida. Tanto a biblioteca de mapas da Google como o OpenStreetMaps, permitem a visualização do mapa em modo offline ou seja, não precisa qualquer ligação à internet. No entanto, essa visualização offline é limitada para a biblioteca da Google, uma vez que tem que ser o utilizador a definir manualmente uma área do mapa, guardando-a na cache do dispositivo móvel. A biblioteca OpenStreetMaps é portanto, a solução mais adequada para a utilização de mapas na aplicação móvel desenvolvida, uma vez que não tem limitações na utilização de mapas sem requerer uma ligação á internet.

No que diz respeito á representação de dados geográficos num browser, a biblioteca PolyMaps parece ser a mais adequada para proceder à representação dos dados. Com recurso a esta ferramenta, somos capazes de representar e visualizar dados, tais como dados populacionais e estatísticos. Parece ser a ferramenta mais adequada para a representação de dados qualitativos e quantitativos.

2.5 Representação de dados estatísticos em mapas

Uma das componentes desenvolvidas no decorrer do estágio consistiu num sistema para visualização de estatísticas sobre o Jardim Botânico, num mapa, apresentando estatísticas à partir de dados recolhidos através da aplicação móvel. Aplicações para a visualização de informação georreferenciada sobre um mapa são essenciais para ajudar um utilizador a obter informação estatística. Estas aplicações requerem uma interface amigável que permite a pesquisa interativa de informação. Além disso, a imagem obtida deve ser inteligível permitindo ao utilizador obter interactivamente mais detalhes [18]. Na cartografia, os mapas tem características específicas que os classificam, e representam elementos seleccionados de um determinado espaço geográfico, de forma reduzida, utilizando simbologia e projeção cartográfica. A elaboração de mapas temáticos abrange as seguintes etapas: recolha de dados, análise, interpretação e representação da informação num mapa.

Destacam-se três tipos de dados: qualitativos, ordenados e quantitativos. O nível de organização dos dados, sejam eles qualitativos, ordenados ou quantitativos, num mapa, está diretamente relacionado com o método de mapeamento e com a utilização de variáveis visuais adequadas à sua representação [20]. A Figura 16 identifica as variáveis

visuais, adequadas para a representação de dados num mapa. Os símbolos, para representação dos dados, são construídos basicamente com a variação visual de forma, tamanho, orientação, cor, valor e granulação para representar fenómenos qualitativos, ordenados ou quantitativos nos modos de implantação pontual, linear ou zonal.

O variável visual *tamanho* corresponde à variação do tamanho do ponto, de acordo com a informação quantitativa; a variável visual *valor* pressupõe a variação da tonalidade ou de uma sequência monocromática; a *granulação* corresponde a variação da repartição do preto no branco onde deve-se manter a mesma proporção de preto e de branco; a variável visual *cor* significa a variação das cores do arco-íris, sem variação de tonalidade, tendo as cores a mesma intensidade. Por exemplo: usar azul, vermelho e verde é usar a variável visual “cor”. O uso do azul-claro, azul médio e azul escuro corresponde à variável “valor”. A variável visual “orientação” corresponde às variações de posição entre o vertical, o oblíquo e o horizontal e, por fim, a forma, agrupa todas as variações geométricas ou não [71].

A combinação das variáveis visuais dará origem aos diferentes tipos de mapas temáticos, entre os quais os mapas de símbolos pontuais, mapas de isolinhas e mapas de fluxos, mapas zonais ou coropléticos, mapas de símbolos proporcionais ou círculos proporcionais, mapas de pontos ou de nuvens de pontos [20]. A seguir, são apresentados os fenómenos quantitativos a qualitativos, sendo estes os dois tipos de fenómenos que são representados no sistema de visualização de estatísticas. São apresentadas as características essenciais para uma correta representação destes tipos de fenómenos, tais como variações na cor, tamanho, entre outros.

Implantation	Pontual	Linear	Zonal
Forma			
Tamanho			
Orientação			
Cor	<p>Uso das cores puras do espectro ou de suas combinações. Combinação das três cores primárias cian, amarelo, magenta (tricomia).</p>		
Valor			
Granulação			

Valor da percepção
 ≡ associativa ≠ seletiva ○ ordenada ◊ quantitativa

Figura 16: Quadro de variáveis visuais [20]

2.5.1 Fenómenos Qualitativos

Os métodos de mapeamento para os fenómenos qualitativos utilizam as variáveis visuais seletivas forma, orientação e cor, nos três modos de implementação: pontual, linear e zonal [20]. A construção de mapas de símbolos pontuais nominais leva em conta os dados absolutos que são localizados como pontos e utiliza como variável visual a forma, a orientação ou a cor. Também é possível utilizar símbolos geométricos associados ou não às cores.

A Figura 17 exemplifica a representação de dados qualitativos. É utilizada uma diferença na cor, consoante se trate de uma área com/sem favelas.

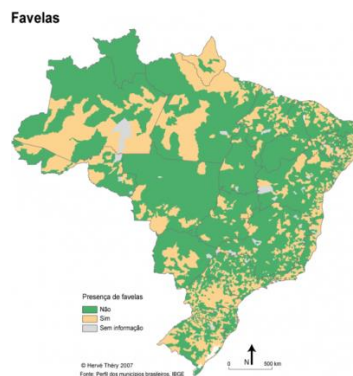


Figura 17: Exemplificação da representação de fenômenos qualitativos [20]

2.5.2 Fenômenos Quantitativos

Os fenômenos quantitativos são representados pelas variáveis visuais “tamanho” e luminosidade, e podem ser implementados em localizações pontuais do mapa ou em implementação zonal, por meio de pontos agregados, como também, na implementação linear com variação da espessura da linha. Os mapas de símbolos proporcionais representam melhor os fenômenos quantitativos e constituem-se num dos métodos mais empregados na construção de mapas com implementação pontual [20]. Esses tipos de mapas são utilizados para representar dados absolutos tais como população em número de habitantes em pontos selecionados do mapa. Geralmente, utiliza-se o círculo proporcional aos valores que cada unidade apresenta em relação a uma determinada variável, porém podem ser utilizados quadrados e triângulos.

Existem essencialmente três maneiras de representar informação quantitativa num mapa: variação na intensidade da cor, variação no tamanho, ou ambos. A Figura 18 exemplifica a representação de fenômenos quantitativos. Pode-se observar que para a representação dos dados, é mantida a mesma cor, no entanto o diâmetro dos círculos varia, consoante a quantidade maior ou menor a representar.

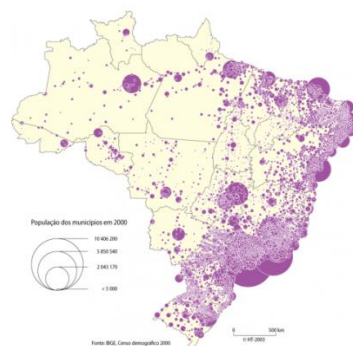


Figura 18: Exemplificação da representação de fenômenos quantitativos [20]

2.5.3 Variação na intensidade da cor

Existem algumas regras de ouro [19] básicas que são úteis na escolha de cores para exibição de dados. É necessário usar:

- Tonalidade pra diferenças qualitativas;
- Luminosidade para diferenças quantitativas (mais escura representa quantidades maiores);
- Variações de saturação num símbolo redundante;
- Grande saturação para pequenas, mas importantes componentes;
- Cores mais iluminadas e menos saturadas para áreas grandes.



Figura 19: Exemplificação da utilização de cores na representação de dados em mapas

As cores representadas na Figura 19 representam um bom modelo de cores que poderia ser aplicado na representação de dados quantitativos num mapa, uma vez que não variam muito pela cor em si, mas pela intensidade da luz, assim como pela saturação que passa de um nível mais claro para um mais escuro, percebendo muito bem como foi ordenado.

Como todas as orientações, existem sempre exceções. Estas orientações abrangem apenas um número limitado de considerações que entram nas opções de cores. No entanto, estas considerações são consideradas um bom ponto de partida. A escolha de cores é sempre um desafio quando se desenha um mapa. Um aspeto importante é não criar uma espécie de “arco-íris”, tentar manter uma cor variando a intensidade da mesma.

2.6 Realidade aumentada

A realidade aumentada (RA) (*augmented reality*) é uma tecnologia que combina elementos do mundo real com elementos virtuais em 3D, permitindo a interatividade entre objetos (reais e virtuais) em tempo real [42]. A informação não segue apenas a pessoa, como também o seu olhar: olhar para um objeto é suficiente para recuperar e exibir informação relevante, ampliando a sua inteligência [22]. Embora a investigação sobre RA tenha avançado nas últimas décadas, a tecnologia de RA ainda tem que chegar ao mercado de massa. O requisito mínimo para conseguir realidade aumentada é ter um dispositivo, uma câmara e uma unidade de processamento. Navegação na *web*, multimédia e reprodução digital de fotografia, são apenas algumas das suas capacidades dos telefones com câmara. Utilizar realidade aumentada em *smartphones* consiste em tornar

disponíveis essas capacidades para milhões de utilizadores e, além disso, fornece uma rápida plataforma de pesquisa. Investigação sobre RA combina com computação móvel e ubíqua em geral assim como com interação 3D. Em computação ubíqua, o computador torna-se “invisível”. Em realidade aumentada, o computador é transparente e o utilizador percebe o mundo através deste. Isto significa que um computador pode misturar impressões do mundo real com informações geradas pelo computador, aumentando desta forma a realidade. O como, como sendo tridimensional e interativo, requer um sistema de realidade aumentada com as seguintes três características: combinar o real com o virtual, interativo em tempo real, registado em 3D.

Existe uma vasta gama de áreas para aplicações de realidade aumentada. Para além da medicina e paleontologia, realidade aumentada em jogos é uma extensão lógica para jogos [69] [70]. Realidade aumentada também estende-se cada vez mais aos serviços baseados na localização. Aplicações de realidade aumentada podem ser usadas em museus, galerias, ou espaços abertos (i.e. Jardins) onde objetos como quadros, esculturas, artefactos, ruínas, plantas entre outros, podem ser “aumentados” com informação (textos, imagens, sons ou vídeos) que será transmitida aos utilizadores, através de um dispositivo, assim que eles desejam recebe-la ou se aproximam do objeto [24]. Um exemplo de uma aplicação deste tipo é o *Ubiwhere Multimedia Museum Guide*, um guia turístico para museus desenvolvido pela empresa Ubiwhere. Mais pormenores sobre esta aplicação podem ser encontrados na secção 2.2.2.

2.6.1 Realidade aumentada aplicada ao turismo

Uma das áreas onde a realidade aumentada é de grande valia, é no turismo. Recorrendo a esta tecnologia, na área em questão, o utilizador pode obter informações, sobre um determinado local, através de informações virtuais, bem como imagens virtuais (tridimensionais) em construções que foram destruídas, ou a guerras acontecidas no passado, bem como em jardins ou parques naturais [40].

Com o auxílio dos sistemas de realidade aumentada, os utilizadores poderão ter acesso a diversas informações uteis e necessárias para o enriquecimento da sua cultura. É possível dizer então que a realidade aumentada proporciona uma melhoria na percepção e interação na combinação do mundo virtual com o mundo real.

2.6.2 Testes a plataformas de realidade aumentada

Segue-se uma lista de plataformas testadas para realidade aumentada. As seguintes plataformas foram analisadas tendo em vista a escolha da mais adequada para integração de uma componente de realidade aumentada na aplicação desenvolvida durante o estágio. Foi feito um estudo demorado, aprofundando as ferramentas que pareciam ser adequadas para utilizar no projeto, de modo a verificar qual delas oferece as condições necessárias para implementar a componente de realidade aumentada na aplicação móvel

Qualcomm: Trata-se de uma plataforma muito poderosa para realidade aumentada. Esta plataforma utiliza a *Vuforia SDK*, uma biblioteca para realidade aumentada, e um sistema

de gestão alvo, hospedado no portal de programadores [25]. A *Vuforia SDK* usa o ecrã do dispositivo móvel como uma “lente mágica” num mundo aumentado, onde os mundos reais e virtuais podem coexistir. A aplicação processa a imagem de visualização ao vivo. Objetos 3D virtuais são então sobrepostos à visualização da câmara ao vivo e parecem estar intimamente ligados no mundo real. Esta plataforma faz o uso da biblioteca OpenGL para a representação dos objetos 3D.

A grande desvantagem desta plataforma é a de não permitir georreferenciação dos dados. *Vuforia* utiliza “*Computer vision*” para detetar e mapear a posição e orientação das imagens que são predefinidas como “*targets*”. O SDK não usa GPS ou outros serviços de localização. A Figura 20 representa um teste efetuado pelo autor com a biblioteca Qualcomm.



Figura 20: Representação de um teste efetuado com a plataforma Qualcomm

Layar: Trata-se de uma plataforma de realidade aumentada de tipo Browser. Esta plataforma é especializada em geolocalização e realidade aumentada. Recorrendo a esta ferramenta, é possível mostrar ao utilizador informação presente ao seu redor, utilizando a câmara e o GPS do dispositivo móvel. Para a inserção de pontos de interesse na aplicação, o utilizador deverá aceder à página web do *Layar*, registar-se como programador e adicionar um novo conjunto de dados. Um dos inconvenientes desta plataforma é o facto de apenas funcionar *online*, tratando-se de um browser [37].

Junaio: Trata-se de uma plataforma de realidade aumentada que funciona para os sistemas operativos Android e iOS. É uma plataforma baseada na localização, mostrando os pontos de interesses nas proximidades do utilizador, ligando-se a plataformas como Foursquare, Instagram, Ebay, entre outros [37]. Para além disso, também é possível visualizar modelos 3D. É portanto uma boa ferramenta para geolocalização como para reconhecimento de imagens [26]. A principal desvantagem desta plataforma é que necessita uma conexão permanente à internet, de modo a conseguir obter os conteúdos a apresentar aos utilizadores. Para criar pontos de interesse para esta aplicação, é necessário registar-se como programador no site-web da plataforma. O programador apenas consegue usufruir de todas as funcionalidades da plataforma utilizando uma versão não-gratuita da ferramenta.

Wikitude: Trata-se de uma aplicação móvel que fornece uma plataforma de realidade aumentada. Fornece uma API para programadores que desejam criar aplicações com realidade aumentada. Suporta os sistemas operativos iOS e Android. Para além de desenvolver novas aplicações, recorrendo à API, esta plataforma também permite acrescentar as funcionalidades base da aplicação, naquela que o programador deseja criar. A plataforma permite visualizar pontos de interesse extraídos do *Twitter*, *Wikipédia*,

Flickr, entre outros. Também permite ao programador inserir os seus próprios pontos de interesse. Para além da georreferenciação dos dados, a plataforma permite a representação de objetos 3D e reconhecimento de imagens [27].

Procedeu-se a testes do SDK da Wikitude que é fornecida para programadores de aplicações com realidade aumentada. Após a instalação e configuração do SDK, procedeu-se à realização de alguns testes da plataforma. O *download* do SDK fornece uma pasta com alguns exemplos para importar para o ambiente de desenvolvimento, compilar e testar no *smartphone*. Foi importada uma aplicação que permite visualizar pontos de interesse, criados num perímetro de alguns metros à volta da posição do utilizador. A Figura 21 mostra o resultado obtido.

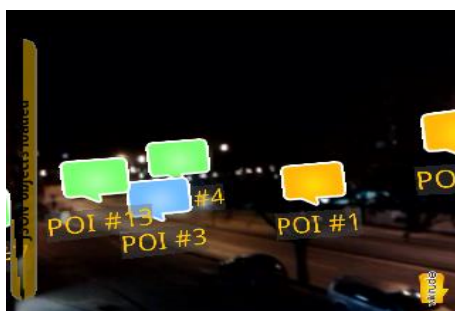


Figura 21: Representação de um teste efetuado pela plataforma Wikitude

Após a realização deste teste, foi feita uma pequena investigação ao código fonte da aplicação testada. Verificou-se que os pontos de interesse são adicionados manualmente pelo programador, não necessitando da utilização de um browser para a inserção de conteúdos, ao contrário de muitas outras aplicações. Verificou-se também que a visualização dos objetos não requer uma ligação à internet. No que toca à integração desta plataforma com outras aplicações, verificou-se no *web-site* do SDK da Wikitude que “com o SDK de realidade aumentada da *wikitude*, está-se apenas a alguns cliques de distância de integrar a plataforma mundial líder de realidade aumentada com a sua aplicação.” Para a representação dos dados são utilizadas tecnologias como HTML5, CSS, javascript, entre outros. No entanto, apesar de todas os benefícios e vantagens desta plataforma, esta é gratuita apenas para fins académicos, apresentando marcas de água na tela do dispositivo móvel. Para poder publicar aplicações com a plataforma wikitude, e desta forma poder remover as marcas de água que aparecem no ecrã, é necessário comprar uma licença.

Mixare: Esta plataforma de realidade aumentada que permite, por omissão, visualizar informação sobre pontos de interesse, extraídos da Wikipédia. Para além disto, a plataforma permite aos programadores efetuarem modificações de modo a obter a informação desejada não só da Wikipédia mas também noutras fontes de informação definidas pelo próprio programador. A aplicação pode portanto ser personalizada [37]. A aplicação consegue funcionar desligada da rede internet. Uma característica muito importante é que permite georreferenciação de dados. No entanto, verificou-se que esta plataforma apresenta uma qualidade “baixa” na apresentação dos dados e não permite a representação de objetos 3D. A Figura 22 mostra os resultados obtidos com a plataforma Mixare. De todas as plataformas de realidade aumentada analisadas nesta secção, esta é a única que é completamente grátis para utilização em aplicações e publicação.



Figura 22: Representação de um teste efetuado com a plataforma Mixare

A Tabela 7, resume os guias analisados de acordo com um conjunto de características distintivas: funcionamento em modo *offline*, plataformas móveis disponíveis, utilização de elementos 3D, georreferenciação de dados e capacidade de integração com outras aplicações.

Aplicação	Modo Offline	Utilização objetos 3D	Permite georreferenciação de dados	Integração com outras aplicações	Plataforma
Qualcomm	Sim	Sim	Não	Sim	iPhone, Android, Unity
Layar	Não	Sim	Sim	Apenas para iPhone	Android, iPhone
Junaio	Não	Sim	Sim	Sim	iPhone, Android,
Wikitude	Sim	Sim	Sim	Sim	iPhone, Android
Mixare	Sim	Não	Sim	Sim	iPhone, Android

Tabela 7: Revisão das plataformas de realidade aumentada

Após analisar algumas das plataformas de realidade aumentada mais populares do mercado, percebeu-se que algumas delas não poderiam ser utilizadas na aplicação desenvolvida. Dado que inicialmente foi definido que a aplicação teria que funcionar em modo *offline*, descartou-se logo a utilização das plataformas Layer e Junaio, uma vez que estas funcionam por Browser. No que diz respeito à ferramenta Junaio, esta apenas oferece funcionalidades completas numa versão paga da ferramenta de desenvolvimento, tornando-se uma limitação à utilização desta plataforma. A plataforma Qualcomm, que suporta a versão do Android desejada e permite funcionar em modo *offline*, revelava-se inicialmente a melhor hipótese. No entanto, verificou-se que esta plataforma não permite georreferenciação de dados, tornando-se uma limitação importante para a aplicação a desenvolver. Finalmente, resta a plataforma Wikitude, que consegue funcionar desligada da internet, permitindo georreferenciação de dados e integração com outras aplicações Android. Esta última plataforma parecia cumprir os requisitos necessários para a sua utilização no projeto. No entanto, após alguns testes, verificou-se que a versão grátis apresenta marcas de água no ecrã do dispositivo móvel, tornando-se um problema de

usabilidade e design para a aplicação. Foi portanto escolhida a plataforma Mixare, apesar de ser mais limitada graficamente em relação á plataforma Wikitude.

2.7 Integração com redes sociais

As redes sociais estão cada vez mais presentes na vida das pessoas, tornando-se uma técnica chave na sociologia moderna. Recorrendo às redes sociais, um utilizador pública diversa informação sobre os seus hábitos, atividades ou gostos. Partilhando essa informação, é possível a outros utilizadores efetuarem pesquisas sobre dados dos utilizadores, para por exemplo tratamento estatístico. Hoje em dia, algumas redes sociais disponibilizam bibliotecas para obter informação sobre utilizadores, páginas ou eventos. No presente tópico, foram analisadas ferramentas para pesquisa de dados nas redes sociais. São apresentadas as ferramentas que foram encontradas, para as redes sociais Facebook,⁵ Twitter⁶ e Google+.

Facebook Graph API: A rede social Facebook disponibiliza uma ferramenta para efetuar pesquisas, tornando-se a principal forma de obter dados dentro e fora do grafo social do Facebook. Trata-se de uma API com base em HTTP, que permite efetuar consulta de dados, postar novas histórias ou efetuar qualquer outra operação que uma aplicação deste tipo pode fazer [53]. Trata-se de um grafo social no sentido da ciência da computação. Não se trata de uma simples tabela de dados. A Graph API apresenta uma visão simples e consistente do grafo social do Facebook, representando uniformemente os objetos no grafo (por exemplo, pessoas, fotos, eventos e páginas) e as conexões entre eles (por exemplo, os relacionamentos entre amigos, conteúdo partilhado, marcadores de fotos). Cada objeto no grafo tem uma identificação única. É possível aceder às propriedades de um objeto, solicitando https://graph.facebook.com/ID_Objeto -, onde ID_Objeto representa o identificador do objeto. As respostas das solicitações são devolvidas no formato JSON. É aqui importante referir que a informação do grafo apenas pode ser consultada em perfis públicos. Caso o perfil do utilizador esteja definido como privado, não é possível aceder à informação pretendida. A Figura 23 apresenta um exemplo da utilização da Graph Api. Um utilizador disponibilizou o identificador da sua conta, assim como o identificador de uma publicação efetuada no seu perfil. Combinando os dois identificadores, é possível obter os comentários associados à publicação em causa.

⁵ [Www.facebook.com](http://www.facebook.com)

⁶ [Https://twitter.com](https://twitter.com)


```

{
  "data": [
    {
      "id": "153125724720582_184234384932460_2455672",
      "from": {
        "name": "Yudha Febrineas",
        "id": "100000275526455"
      },
      "message": "Egi jg, vania k\u00f4nser dimana ni",
      "can_remove": false,
      "created_time": "2011-01-23T02:32:01+0000",
      "like_count": 1,
      "user_likes": false
    },
    {
      "id": "153125724720582_184234384932460_2455686",
      "from": {
        "name": "Leman Loveganza",
        "id": "100001832133158"
      },
      "message": "AMIEH. .\r\nI LOVE YOU VANIA LARISSA JADILAH YG LBH BAIK LG. .\r\nFIGHTING! :)",
      "can_remove": false,
      "created_time": "2011-01-23T02:34:26+0000",
      "like_count": 0,
      "user_likes": false
    },
    {
      "id": "153125724720582_184234384932460_2455697",
      "from": {
        "name": "Artemio E Pascual",
        "id": "684019566"
      },
      "message": "No worries!",
      "can_remove": false,
      "created_time": "2011-01-23T02:36:23+0000",
      "like_count": 0,
      "user_likes": false
    }
  ],
}

```

Figura 23: Exemplo da utilização de Graph API

Toda a informação recolhida, recorrendo à API pode ser de uma grande utilidade para tratamento estatístico. Com isto, é possível determinar por exemplo preferências dos utilizadores.

Twitter Search API: A rede social Twitter disponibiliza uma API dedicada para a realização de pesquisas sobre *tweets* efetuados recentemente por um utilizador. Recorrendo a esta biblioteca, é possível pesquisar “*tweets*” por palavra-chave ou categoria. Tal como a *Graph API* do facebook, esta funciona efetuando simples pedidos HTTP. A API vai na versão 1.1 e, tal como para a *Graph API*, realiza pedidos REST, devolvendo as respostas no formato JSON [60].

Google Social Graph API: A rede social do Google disponibilizava uma API dedicada á pesquisa de informações na rede social. No entanto, A API já não se encontra disponível [64].

A tabela 8 apresenta e compara características das três ferramentas analisadas.

Característica	Facebook Graph API	Twitter Search API	Google Social Graph API
Disponível	Sim	Sim	Não
Pedidos HTTP	Sim	Sim	Sim
Versão		V. 1.1	N.A
Requer Autenticação	Não (apenas fornece dados públicos)	Não (apenas fornece dados públicos)	Não (apenas para dados públicos)
Formatos de Resposta	JSON	Json Atom	JSON

Tabela 8: Comparação de características para as ferramentas analisadas

2.8 Soluções para criação de API REST em Django

Nesta secção são analisadas as bibliotecas existentes, mais populares, para a criação de API's REST utilizando a *framework* Django. A escolha da análise para a *framework* Django deve-se essencialmente ao facto de a empresa usar Django, definindo desde o início do projeto a utilização do mesmo para a criação do servidor. São analisadas três bibliotecas: *django-tastypie*, *django-rest-framework* e *django-piston*⁷.

Django-tasypie: Esta biblioteca permite a criação de API's para a *framework* Django. Oferece aos programadores uma documentação muito boa e completa. Suporta autenticação por OAuth.

Django-rest-framework: Esta biblioteca é muito semelhante à *Tastypie*. Permite a criação de API's REST para Django. Em termos de segurança, oferece já suporte para autenticação com OAuth2. Para além disso, permite a sua utilização em versões de Python 3. A comunidade é bastante ativa e possui uma boa documentação.

Django-piston: Das três bibliotecas aqui analisadas, esta é considerada pela comunidade de programadores como sendo a menos adequada. Esta biblioteca não é atualizada há mais de três anos e possui uma documentação muito fraca. Muitas vezes o código da plataforma não é testado. A única razão pelo uso desta biblioteca durante muitos anos, no passado, é o suporte oferecido para OAuth, no entanto, *django-tastypie* e *Django-rest-framework* também oferecem essa funcionalidade.

A tabela 9 resume e compara as características das três API's apresentadas.

Package	Piston	Django Tastypie	Django Rest Framework
Descrição	Piston é uma mini-framework Django para criação de API's	Tastypie permite criar API's para Django desde 2010	Criação de API's para Django
Categoria	App	App	App
Ultima Atualização	31 Março 2012	19 de Março de 2013	27 de Setembro de 2012
Versão	0.2.3	0.9.15	2.3.5
Interface de Administração	não	Não	Não
Web Browsable API	não	não	sim
Serializations	XML JSON	JSON JSONp	JSON JSON

⁷ <http://pydanny.com/choosing-an-api-framework-for-django.html>

	YAML Pickle Django	XML YAML HTML	XML YAML HTML
Accept Headers	Não	Sim	sim
API key authentication	Não	Sim	sim
OAuth	Sim	Sim	Sim
OAuth2	Não	Não	Sim
Python 3	Não	Não	Sim

Tabela 9: Comparação de características para as ferramentas de criação de APIs REST em Django ⁸

A análise às três *framework* para criação de API's REST permite concluir que a *framework* Piston não é adequada para a criação de API's. Tratando-se de uma *framework* que foi utilizada durante muito tempo pelos programadores, esta tornou-se inadequada com o aparecimento de problemas de segurança, assim como com o aparecimento de outras *frameworks* mais seguras e que oferecem melhor documentação. Para além disso, Piston não é atualizada há algum tempo parecendo portanto ser um projeto eu já não parece ser importante para a comunidade. A utilização da *framework* Tastypie ou Django Rest Framework torna-se mais adequada. No entanto, dado que a API da empresa utiliza Tastypie, será esta a *framework* a ser utilizada.

2.9 Soluções para utilização de bases de dados

Este secção pretende analisar características de alguns motores de bases de dados. Com isto, pretende-se efetuar uma análise tendo em vista a escolha do motor mais adequado para ser utilizado na parte do servidor, para armazenamento dos dados. Foram analisados os motores SQL Server, Oracle, MySQL e PostgreSQL. De seguida, é apresentada uma tabela que resume e compara características dos motores analisados.

Característica	SQL Server	Oracle	MySQL	PostgreSQL
Open Source			X	X
Free / No License Costs			X	X
ACID compliant	X	X	X	X
ANSI SQL Compliant	X	X		X
Referential Integrity	X	X	X	X
Replication	X	X	X	X
Rules	X	X		X
Views	X	X	X	X
Triggers	X	X	X	X

⁸ <https://www.djangopackages.com/grids/g/api/>

Unicode	X	X		X
Sequences		X	X	X
Inheritance		X		X
Outer Joins	X	X	X	X
Sub-selects	X	X	X	X
Open API			X	X
Stored Procedures	X	X	X	X
Native SSL Support	X	X	X	X
Procedural Languages	X	X		X
Indexes	X	X	X	X

Tabela 10: Comparação de características para os motores de bases de dados [32]

No que toca ao suporte para objetos geográficos, o motor de base de dados PostgreSQL utiliza o PostGIS[31], uma extensão que oferece suporte para objetos geográficos. PostGIS permite que o motor de base de dados seja usado como uma base de dados de *back-end* espacial para sistemas de informação geográfica (GIS). Outro fator importante é que PostgreSQL é um motor de base de dados de baixo custo, tendo uma comunidade de suporte *open source*. A Oracle oferece como alternativa o *Oracle Spatial*. No entanto, trata-se de uma extensão que não é grátis, ao contrário do PostGIS que é grátis, oferecendo menos custos em relação à Oracle [33]. Portanto olhando para todos os elementos apresentados é possível concluir que o motor de base de dados PostgreSQL foi uma boa escolha para ser utilizado no projeto. .

Capítulo 3

Análise dos Requisitos

Apresenta-se neste capítulo a análise efetuada para a definição dos requisitos da aplicação desenvolvida. A especificação dos requisitos da aplicação foi feita recorrendo à definição de *user stories* e especificação de casos de uso. Uma *User story* é uma definição de muito alto nível de um requisito, contendo apenas a informação suficiente para que os desenvolvedores consigam produzir uma estimativa razoável do esforço a produzir para implementá-lo [54]. Um caso de uso consiste numa sequência de ações que proporcionam um valor mensurável para um ator. Um caso de Uso descreve a maneira pela qual um ator do mundo real interage com o sistema [55]. Neste capítulo são analisados os requisitos funcionais e não-funcionais da aplicação, seguindo um formato especificado a seguir no documento, assim como os atores intervenientes no sistema.

3.1 Atores

Os atores intervenientes no sistema são o Visitante do Jardim e a entidade gestora dos conteúdos do Jardim. O ator Visitante do Jardim tem como principal responsabilidade a utilização da aplicação, com recurso a um *smartphone* Android quando efetuar a visita ao Jardim Botânico. O ator entidade gestora dos conteúdos do Jardim tem como responsabilidade a gestão dos conteúdos presentes na aplicação. Esta entidade tem acesso a uma interface que permite adicionar, alterar ou eliminar conteúdos presentes no *back-office* da aplicação.

3.2 Requisitos funcionais do sistema

Esta secção identifica os requisitos funcionais do sistema. São primeiro identificadas as *user stories*. Numa segunda fase, com apoio das *user stories*, são descritos os casos de uso.

3.2.1 Definição de *user stories*

As *user stories* respondem a três questões importantes na definição das funcionalidades a implementar:

- O que se pretende?
- Quem o pretende?
- Porquê?

Definir o ator implica pensar no interveniente que utilizará a funcionalidade. A ação descreve o que acontecerá mas não como esta deverá ser executada. Portanto, para a definição das *user stories* será utilizado o seguinte *template*:

- Enquanto <utilizador>
- Posso <ação>

- De forma a <proposta de valor>

As *user stories* foram classificadas em várias categorias, dependendo da funcionalidade que está por detrás de cada uma delas. Segue-se uma listagem das *user stories* agrupadas então por categorias.

Instalação e Updates

US-1 – Enquanto visitante, posso descarregar a aplicação e instalá-la no meu *smartphone* de forma a efetuar a visita guiada no jardim.

Acesso à aplicação

US-2 – Enquanto utilizador, posso entrar na aplicação de forma a ter acesso aos conteúdos disponibilizados pela mesma.

Posicionamento e navegação

US-3 - Enquanto utilizador, posso aceder ao mapa do jardim e ver nele o meu posicionamento atual, de forma a contextualizar a minha posição no interior do jardim. Para isso, o dispositivo GPS do *smartphone* deverá estar ligado.

US-4 - Enquanto utilizador, posso aceder ao mapa do jardim de forma a ver as plantas que se encontram próximas da minha posição.

US-5 – Enquanto visitante ligado à aplicação, posso escolher seguir uma rota pré-definida disponível na aplicação, de forma a tornar mais agradável a minha experiência durante a visita ao jardim.

US – 6 – Enquanto utilizador, posso escolher criar a minha própria rota e segui-la, escolhendo os pontos(plantas) que desejo visitar ao longo do percurso.

US – 7 – Enquanto utilizador, posso alterar a ordem de visita dos pontos da rota que eu criei.

US – 8 - Enquanto utilizador, posso escolher seguir uma rota através da realidade aumentada ou então através da utilização de um mapa 2D.

US – 9 – Enquanto utilizador, posso escolher procurar por uma determinada planta no interior do Jardim, através da utilização da realidade aumentada ou então através da utilização de um mapa 2D.

Visualização de informação

US-10 – Enquanto utilizador, posso aceder a uma listagem das plantas existentes no interior do jardim.

US-11 – Enquanto utilizador, posso alterar a ordem da listagem, de acordo com a minha posição atual.

US-12 - Enquanto utilizador, posso alterar a ordem da listagem, fornecendo os dados por ordem alfabética.

US-13 – Enquanto utilizador, posso alterar a ordem da listagem das plantas, apresentando os dados pela sua relevância.

US-14 – Enquanto utilizador, posso alterar a ordem da listagem das plantas, apresentando os dados pelo número de *likes* associado a cada planta.

US-15 – Enquanto utilizador, posso alterar a ordem da listagem das plantas, apresentando os dados pelo número de visualização de uma planta feitas através da aplicação móvel.

US-16 – Enquanto utilizador, posso alterar a forma de listagem das plantas, visualizando-as numa grelha ou numa lista.

US-17 – Enquanto utilizador, posso aceder à descrição de uma planta selecionada, de forma a conhecer mais detalhes acerca da planta.

US-18 – Enquanto utilizador, posso aceder a vários conteúdos multimédia sobre cada planta presente no jardim.

US-19 – Enquanto visitante ligado à aplicação, posso utilizar a câmara do dispositivo móvel, de modo a visualizar objetos 3D (plantas) recorrendo à realidade aumentada.

US-20 – Enquanto visitante ligado á aplicação, posso utilizar a câmara do dispositivo móvel, de modo a visualizar dados sobre localização e posicionamento, recorrendo à realidade aumentada.

Pesquisa

US-21 – Enquanto utilizador, posso efetuar uma pesquisa sobre uma planta através de uma barra de pesquisa de forma a aceder à informação detalhada da mesma.

US-22 – Enquanto utilizador, posso pesquisar e visualizar rotas pré-definidas pela aplicação.

US-23 – Enquanto utilizador, posso pesquisar e visualizar rotas criadas por mim próprio.

Partilha em redes sociais

US-24 – Enquanto utilizador com conta registada no *Facebook* posso partilhar informação de uma planta nesta rede social.

US-25 – Enquanto utilizador com conta registada no *Twitter* posso partilhar informação de uma planta nesta rede social.

Visualização de dados estatísticos

US-26 – Enquanto entidade gestora do Jardim Botânico, posso aceder a uma página *web*, recorrendo a um browser, de forma a visualizar vários dados estatísticos recolhidos pela aplicação móvel.

US-27 – Enquanto entidade gestora do Jardim Botânico, posso visualizar estatísticas sobre as plantas mais/menos visualizadas no Jardim.

US-28 – Enquanto entidade gestora do Jardim Botânico, posso visualizar estatísticas sobre as plantas que mais/menos “likes” recolhem.

US-29 – Enquanto entidade gestora do Jardim Botânico, posso visualizar estatísticas sobre as plantas que são mais/menos partilhadas em redes sociais.

US-30 – Enquanto entidade gestora do Jardim Botânico, posso visualizar estatísticas sobre as plantas que reúnem mais comentários nos *posts* associados nas redes sociais.

US-31 – Enquanto entidade gestora do Jardim Botânico, posso visualizar estatísticas sobre as zonas do Jardim Botânico mais/menos frequentadas.

Gestão de conteúdos

US-32 – Enquanto entidade gestora do Jardim Botânico, posso aceder à página de gestão de conteúdos do servidor, de forma a inserir elementos presentes no servidor.

US – 33 - Enquanto entidade gestora do Jardim Botânico, posso aceder à página de gestão de conteúdos do servidor, de forma a remover elementos presentes no servidor.

US – 34 - Enquanto entidade gestora do Jardim Botânico, posso aceder à página de gestão de conteúdos do servidor, de forma a alterar elementos presentes no servidor.

3.2.2 Modelo de casos de utilização

De maneira a completar a definição dos requisitos, foram elaborados modelos de casos de uso. Para a definição dos requisitos foi utilizado o seguinte *template*:

Componente	Significado
Tipo	O tipo de requisito. Mais abaixo são apresentados os tipos de requisitos
Prioridade	O grau de prioridade dos requisitos. Mais abaixo são apresentados os tipos de prioridades aplicadas.
Descrição	uma descrição geral do requisito, aquilo que será possível realizar com a implementação do requisito.
Atores	Os atores intervenientes
Pré-condições	condições necessárias para a realização da ação.
Fluxo de trabalho	uma listagem das ações a realizar por parte do ator de modo a chegar ao resultado pretendido.
Resultados esperados	uma listagem dos resultados esperados.

Tabela 11: Componentes do *template* da definição dos requisitos

Priorização dos requisitos

Os requisitos devem seguir normas de garantias de qualidade e aplicabilidade, logo foi criada uma escala de priorização para que seja mencionada em cada requisito descrito. A tabela 12 apresenta a escala de priorização.

Categoria	Significado
Must	Requisitos essenciais que devem ser obrigatoriamente implementados.
Should	Requisitos importantes que devem ser implementados, mas, se não o forem, o sistema poderá ser implantado e usado normalmente.
Could	Requisitos desejáveis que podem ser implementados se não afetarem outros requisitos e o <i>deadline</i> do projeto.
Won't	Requisitos não implementados durante o projeto por questões de recursos/tempo, mas que podem ser implementados no futuro.

Tabela 12: Categorias de priorização de requisitos

Tipos de requisitos

Existem 5 classificações básicas para a descrição dos requisitos que devem ser seguidas para a classificação dos mesmos. A tabela 13 apresenta as categorias possíveis para o tipo de requisitos.

Tipo de Requisito	Significado
Funcionalidade	São as principais funcionalidades da aplicação.
Usabilidade	Determina a estética e coerência na interface com o utilizador.
Confiança	Capacidade da aplicação em recuperação a falhas, disponibilidade e exatidão.
Desempenho	Determina características de velocidade e do tempo de resposta das atividades da aplicação.
Suporte	Determina características como testabilidade, durabilidade, compatibilidade, configurabilidade, instabilidade, escalabilidade e outros.
Segurança	Os requisitos de segurança de software são o conjunto de necessidades de segurança que o software deve atender, Engloba características tais que confidencialidade, integridade, disponibilidade, autenticação, autorização, entre outros.

Tabela 13: Categorias de tipos de requisitos

Segue-se uma listagem dos casos de uso, seguindo o *template* e outras especificações descritas anteriormente. Tal como para as *user stories*, os casos de uso foram categorizados por módulos.

RF-1 – Acesso à localização dentro do recinto do jardim

Tipo: Funcionalidade

Prioridade: *Must*

Descrição: A aplicação deverá permitir ao utilizador consultar a sua localização dentro do jardim através da utilização do GPS do *smartphone*. A localização deverá ser visível num mapa que representa a estrutura do jardim.

Atores: Visitante

Pré-condições: GPS ativo

Fluxo de trabalho:

1. O utilizador escolhe a opção para visualizar a sua posição no mapa do jardim

Resultados esperados:

1. O utilizador visualiza a sua posição no mapa do jardim

RF-2 – Navegação dentro do jardim

Tipo: Funcionalidade

Prioridade: *Must*

Descrição: A aplicação deverá ser capaz de contextualizar o utilizador em termos de posicionamento geográfico dentro do jardim Botânico. Será apresentado um mapa do recinto em questão com a indicação da localização atual do utilizador. À medida que este se vai deslocando a sua posição será atualizada.

Atores: Visitante

Pré-condições: GPS ativo

Fluxo de trabalho:

1. O utilizador acede ao mapa do jardim
2. O utilizador navega pelo jardim podendo visualizar a sua posição pelo mapa

Resultados esperados:

1. O utilizador visualiza a sua posição no mapa à medida que vai circulando pelo jardim.

Módulo de Visualização de informação

RF-3 – Apresentação de rotas pré-definidas ao utilizador

Tipo: Funcionalidade

Prioridade: *Must*

Descrição: A aplicação deverá permitir apresentar ao utilizador um conjunto de rotas pré-definidas que podem ser seguidas dentro do jardim. As rotas deverão apresentar localizações específicas que poderão ser seguidas pelo utilizador formando um percurso.

Atores: Visitante

Pré-condições: O utilizador entrou na aplicação

Fluxo de trabalho:

1. O utilizador seleciona a opção para visualizar as rotas existentes.

Resultados esperados:

1. O utilizador visualiza a lista de rotas pré-definidas existentes

RF-4 – Escolha de uma rota pré-definida

Tipo: Funcionalidade

Prioridade: *Must*

Descrição: A aplicação deverá permitir ao utilizador escolher, caso o deseje, uma rota pré-definida existente na aplicação, permitindo ao utilizador usufruir de uma melhor experiência.

Atores: Visitante

Pré-condições: O utilizador entrou na aplicação

Fluxo de trabalho:

1. O utilizador acede ao menu da aplicação e escolhe a opção de apresentação de rotas
2. O utilizador escolhe uma rota predefinida.

Resultados esperados:

1. O utilizador escolhe uma rota e segue o percurso indicado. A rota poderá ser visualizada através da marcação de pontos específicos representando plantas a visualizar.

RF-5 – Escolha de uma rota criada pelo utilizador**Tipo:** Funcionalidade**Prioridade:** *Must***Descrição:** A aplicação deverá permitir ao utilizador escolher, caso o deseje, uma rotas criadas por ele próprio, podendo escolher a ordem dos pontos a visitar.**Atores:** Visitante**Pré-condições:** O utilizador entrou na aplicação**Fluxo de trabalho:**

1. O utilizador acede ao menu da aplicação e escolhe a opção de apresentação de rotas
2. O utilizador escolhe uma rota criada por ele.
3. O utilizador escolhe a ordem dos pontos para a visita.
4. O utilizador escolhe o modo de visualização da rota (2D ou realidade aumentada)

Resultados esperados:

2. O utilizador escolhe uma rota e segue o percurso em função da ordem que atribuiu aos pontos da rota. A rota poderá ser visualizada através da marcação de pontos específicos representando plantas a visualizar. A rota poderá ser seguida através do mapa 2D ou através da realidade aumentada.

RF-6 – Visualização das plantas que rodeiam o utilizador**Tipo:** Funcionalidade**Prioridade:** *Must***Descrição:** A aplicação deverá permitir ao utilizador visualizar as plantas que o rodeiam a cada instante. Deverá ser apresentada uma imagem representativa por cada planta que se encontra próxima do utilizador.**Atores:** Visitante**Pré-condições:** O utilizador entrou na aplicação**Fluxo de trabalho:**

1. O utilizador acede ao mapa do jardim, visualizando a sua posição
2. O utilizador visualiza no mapa as plantas que o rodeiam.

Resultados esperados:

O utilizador visualiza no mapa as plantas que o rodeiam num determinado perímetro à sua volta.

RF-7 – Visualização de detalhes das plantas através de conteúdo multimédia.

Tipo: Funcionalidade

Prioridade: *Must*

Descrição: A aplicação deverá permitir ao utilizador a visualização de detalhes de uma determinada planta. Caso o utilizador assim o desejar, o guia deverá apresentar para qualquer uma das plantas todos os detalhes da mesma sob a forma de conteúdo multimédia. Caso os conteúdos estejam disponíveis, poderão apresentar-se sobre a forma de texto, fotografias ou vídeos.

Atores: Visitante registado

Pré-condições: o utilizador entrou na aplicação

Fluxo de trabalho:

1. O utilizador acede ao catálogo de plantas.
2. O utilizador escolhe uma das plantas presente no catálogo
3. O utilizador escolhe um dos tipos de multimédia pertencente à planta em questão.

Resultados esperados:

1. O utilizador visualiza o conteúdo multimédia da planta selecionada no catálogo.

RF-8 – Visualização de informação através de realidade aumentada

Tipo: Funcionalidade

Prioridade: *Must*

Descrição: A aplicação deverá permitir ao utilizador navegar no interior do Jardim assim como visualizar informação recorrendo á câmara do dispositivo móvel. O utilizador será capaz de visualizar recursos em “realidade aumentada”. O utilizador poderá por exemplo visualizar as plantas em 3D apontando a câmara do dispositivo móvel em direção às plantas.

Atores: Visitante

Pré-condições: GPS ativo

Fluxo de trabalho:

1. O utilizador escolhe a opção responsável por ligar a câmara;
2. O utilizador aponta a câmara para uma planta.

Resultados esperados:

1. O utilizador visualiza a planta em 3D.

Módulo de Pesquisa

RF-9 – Efetuar pesquisa em todos os conteúdos

Tipo: Usabilidade

Prioridade: *Should*

Descrição: A aplicação deverá permitir ao utilizador efetuar pesquisas de uma planta entre todas as plantas presentes no jardim botânico, através de uma barra de pesquisa, podendo também aceder aos detalhes de cada uma destas.

Atores: Visitante

Pré-condições: O utilizador entrou na aplicação

Fluxo de trabalho:

1. O utilizador acede à barra de pesquisas.
2. O utilizador insere uma pesquisa de planta na barra de pesquisas.
3. O utilizador escolhe a planta desejada

Resultados esperados:

1. O utilizador visualiza os detalhes sobre a planta selecionada

Módulo de Gestão de conteúdos

RF-10 – Sincronização dos conteúdos

Tipo: Suporte

Prioridade: *Must*

Descrição: A aplicação deverá permitir a possibilidade de efetuar a sincronização dos conteúdos (plantas) presentes no interior do jardim assim como de toda a informação e conteúdos pertencentes a cada uma das plantas. Com isto, os conteúdos *offline* presentes na aplicação estão sincronizados com os conteúdos presentes no servidor.

Atores: Utilizador

Pré-condições: O visitante encontra-se num ponto *wireless*

Fluxo de trabalho:

1. O visitante liga-se à internet através, por exemplo, de um ponto *wireless*
2. A aplicação deteta a existência de uma ligação à *internet ativa* e procede à sincronização dos conteúdos.

Resultados esperados:

1. Caso o utilizador tenha uma conexão à internet ligada, a sincronização dos conteúdos é efetuada. A sincronização será feita de maneira transparente para o utilizador. Sempre que é detetada ligação à internet, a aplicação vai verificar automaticamente se os conteúdos estão atualizados.
2. Caso o utilizador não esteja ligado à internet, a sincronização dos conteúdos não é executada.

Módulo de partilha com redes sociais

RF-11 – Partilha de recursos com as redes sociais

Tipo: Funcionalidade

Prioridade: *Must*

Descrição: A aplicação deverá permitir a partilha de recursos pelas principais redes sociais (*Facebook, Twitter*). Com isto, o utilizador poderá partilhar, na rede social desejada, um *post* descrevendo, por exemplo, a planta que visitou no interior do Jardim Botânico.

Pré-condições: O visitante encontra-se ligado à internet e tem conta nas redes sociais respetivas

Fluxo de trabalho:

1. O utilizador visualiza um determinado recurso na aplicação;
2. O utilizador escolhe o botão relativo à rede social desejada;
3. O utilizador confirma a partilha do recurso na rede social.

Resultados esperados:

1. Caso o utilizador esteja ligado à internet, a partilha é efetuada com sucesso na rede social;
2. Caso o utilizador não esteja ligado à internet, a partilha do recurso na rede social não é efetuada.

Módulo de Representação de dados estatísticos

RF-12 – Visualização das métricas recolhidas

Tipo: Funcionalidade

Prioridade: *Must*

Descrição: Será disponibilizada uma aplicação web para permitir visualizar a representação de métricas recolhidas pela aplicação. As métricas a serem representadas são estatísticas representando as zonas mais visitadas do Jardim Botânico, as plantas mais pesquisadas e visualizadas pela aplicação móvel, as plantas mais partilhadas nas redes sociais/que mais comentários recolhem, entre outras mais métricas que podem ser definidas à medida que vai evoluindo o projeto.

Pré-Condições: O utilizador recorre a um *browser* para a visualização dos dados

Fluxo de trabalho:

1. O utilizador acede a um browser e entra no site responsável pela visualização dos dados;
2. O utilizador escolhe uma das métricas possíveis para visualizar, numa lista contendo todas as métricas visualizáveis.

Resultados esperados:

1. O utilizador visualiza os dados relativos à métrica escolhida. A informação é representada num mapa.

3.3 Requisitos não funcionais do sistema

RNF-1 – Compatibilidade

Tipo: Suporte

Prioridade: *Must*

Descrição: A aplicação deverá ser compatível e acessível através de um *smartphone* com o sistema operativo móvel Android. A versão mínima suportada para o sistema operativo Android será 2.2.

RNF-2 – Funcionamento *offline*

Tipo: Confiança

Prioridade: *Must*

Descrição: A aplicação deverá correr em modo *offline*, desligada de qualquer rede internet. O utilizador deverá ser capaz de usufruir de todas as funcionalidades da aplicação, com exceção às funcionalidades envolvendo redes sociais, com a internet desligada. Sempre que o utilizador desejar aceder às redes sociais, utilizando a aplicação, deverá estar ligado à internet.

RNF-3 – Disponibilidade do sistema

Tipo: Confiabilidade

Prioridade: *Must*

Descrição: O sistema deverá ter alta disponibilidade. Considera-se que o sistema deverá ter uma disponibilidade de 99% do tempo.

RNF-4 – Acesso aos dados de um utilizador

Tipo: Segurança

Prioridade: *Must*

Descrição: No que toca à recolha de dados do utilizador, apenas será guardada a sua localização, não necessitando guardar qualquer outro tipo de dados privados pertencente a um utilizador. As coordenadas recolhidas não são portanto associadas a nenhum utilizador em concreto.

RNF-5 – Tolerância a falhas

Tipo: Confiabilidade

Prioridade: *Must*

Descrição: O sistema deverá ser capaz de lidar com falhas de rede aquando a sincronização de dados entre o servidor e a aplicação presente no dispositivo móvel. A aplicação deverá ser capaz de retomar a ligação à rede e obter os conteúdos que não recebeu aquando a falha de rede. Para isso será utilizado um mecanismo baseado em troca de *Tag's* entre servidor e dispositivo móvel, que permite verificar a versão dos dados presentes do lado do servidor e do lado da aplicação móvel. Mais pormenores sobre este mecanismo poderão ser encontrados na secção da arquitetura.

3.4 Diagramas de casos de uso

Nesta secção é apresentado um diagrama de casos de uso de modo a perceber melhor as funcionalidades do produto. O diagrama de casos de uso foi dividido em dois diagramas: um para representar os casos de uso pertencentes á aplicação móvel e um segundo diagrama para os casos de uso do sistema de gestão de conteúdos/visualização de estatísticas.

3.4.1 Diagrama de casos de uso para a aplicação móvel

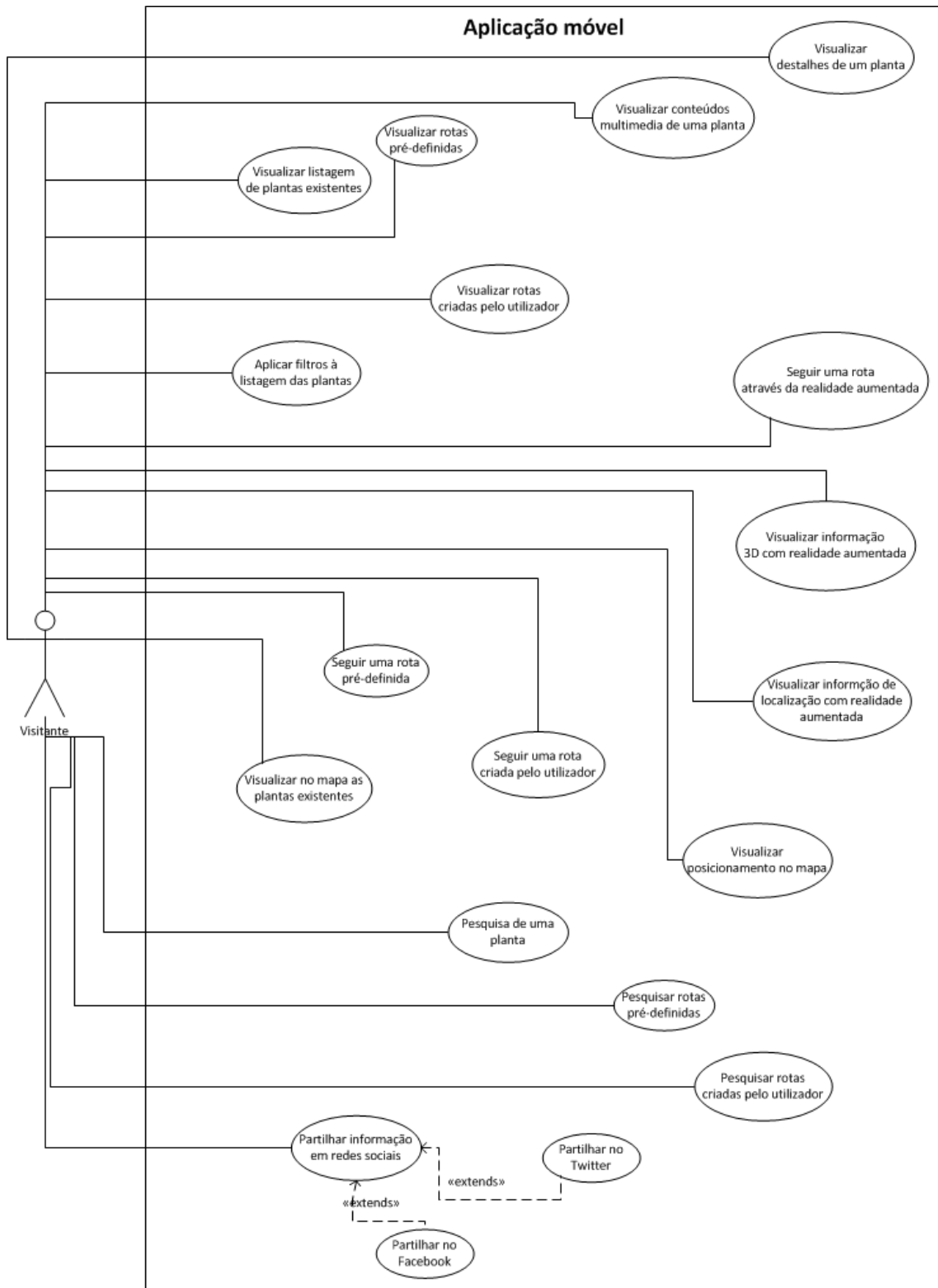


Figura 24: Diagrama de casos de uso da aplicação móvel

3.4.2 Diagrama de casos de uso para a gestão de conteúdos/Visualização de estatísticas

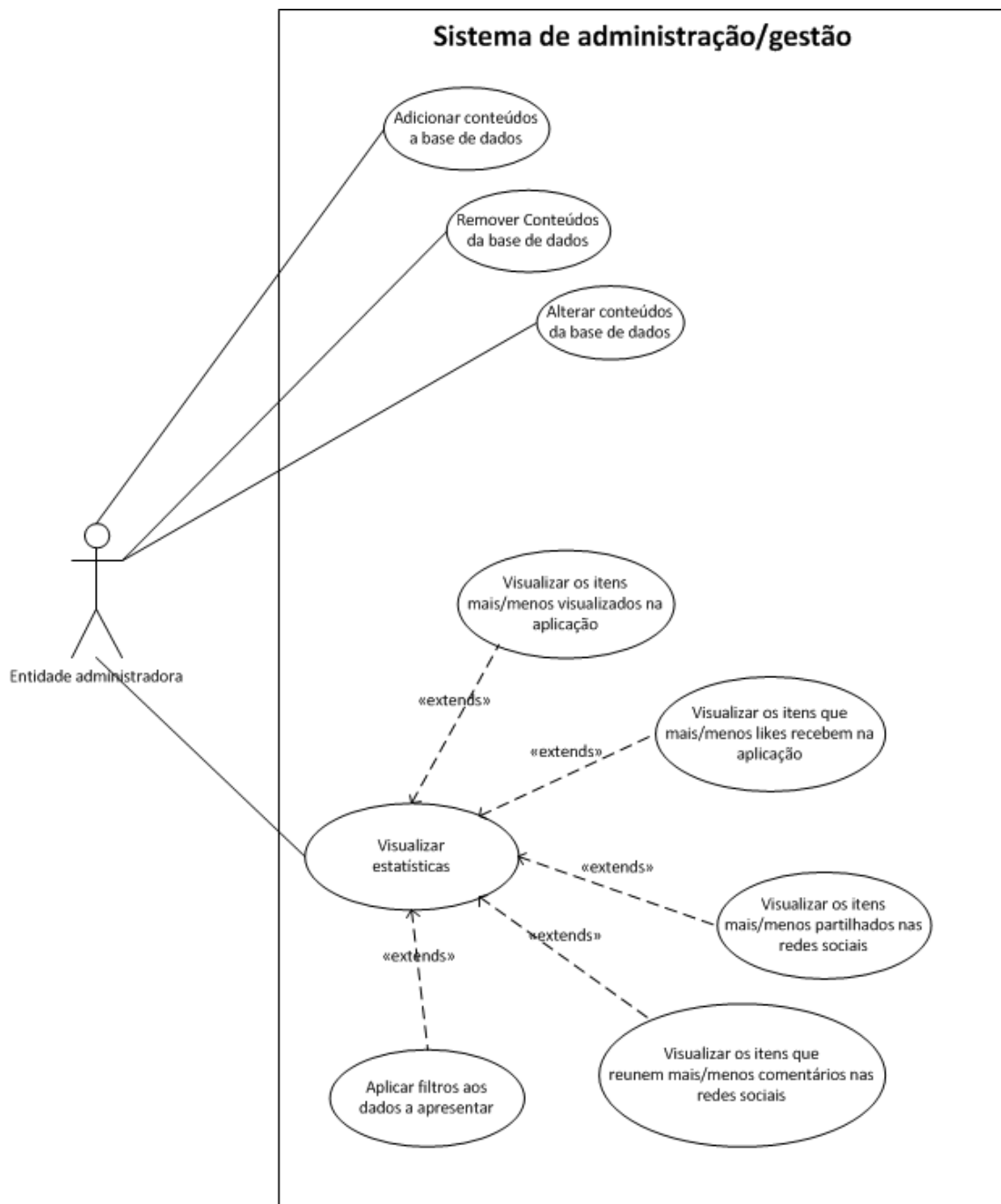


Figura 25: Diagrama de casos de uso para a parte de administração/gestão de conteúdos

3.5 Conclusão

Este capítulo serviu para elaborar os requisitos para a aplicação móvel. Com a elaboração deste documento possui-se uma visão mais clara das funcionalidades a implementar para a aplicação. Para isso recorreu-se a elaboração de *user stories*, requisitos funcionais apoiados por casos de uso, assim como a requisitos não funcionais. Finalmente foram também criados alguns diagramas de casos de usos para perceber mais rapidamente o que a aplicação é capaz de fazer. Resumindo, a aplicação é capaz de fornecer funcionalidade de navegação no interior do jardim, recorrendo ao GPS do dispositivo móvel. A aplicação permite visualizar informações sobre plantas presentes no interior do jardim. É também possível efetuar pesquisas de plantas ou rotas. O Utilizador pode partilhar a sua experiência durante a visita ao Jardim através das redes sociais Facebook e Twitter. Finalmente, é possível visualizar dados no interior do jardim recorrendo à realidade aumentada.

Capítulo 4

Definição da Arquitetura

Neste capítulo é descrita a arquitetura do sistema desenvolvido tendo como objetivo descrever elementos que foram relevantes para a implementação da solução. A definição da arquitetura aparece numa fase de transição entre o levantamento de requisitos do sistema e a implementação do mesmo.

4.1 Vista de alto nível da arquitetura

Este tópico serve para apresentar uma vista de alto nível da arquitetura, de modo a contextualizar o leitor para os próximos tópicos do documento. A arquitetura a desenvolvida segue o paradigma Cliente-Servidor. Existe um cliente, que consiste numa aplicação móvel para *smartphone*, utilizando o sistema operativo Android, e um servidor que consiste numa API REST, trocando dados com a aplicação móvel. As duas componentes trocam dados com pedidos REST, que assenta sobre o protocolo HTTP. A Figura 26 mostra a arquitetura aqui descrita. A arquitetura é descrita mais pormenorizadamente nos seguintes tópicos do documento.

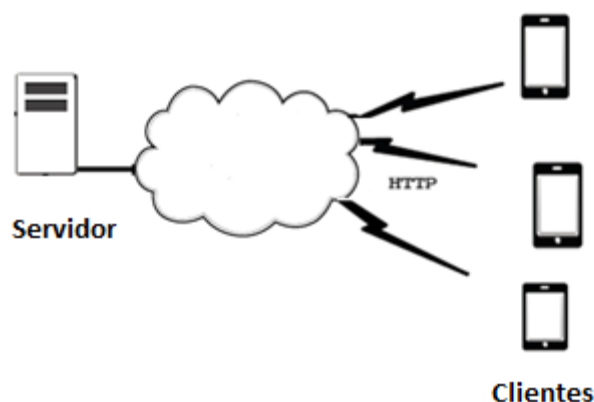


Figura 26: Diagrama de alto nível da arquitetura

4.2 Requisito para a arquitetura

4.2.1 Sistema operativo

Tal como definido no capítulo dos requisitos, a aplicação móvel foi desenvolvida para o sistema operativo Android.

O Android é uma plataforma aberta (OpenSource) de desenvolvimento de aplicações para dispositivos móveis suportada pela Google. O SDK do Android fornece

APIs assim como ferramentas necessárias para o desenvolvimento das aplicações móveis, utilizando para isso a linguagem Java. A Figura 27 apresenta as várias camadas da arquitetura do sistema operativo Android.

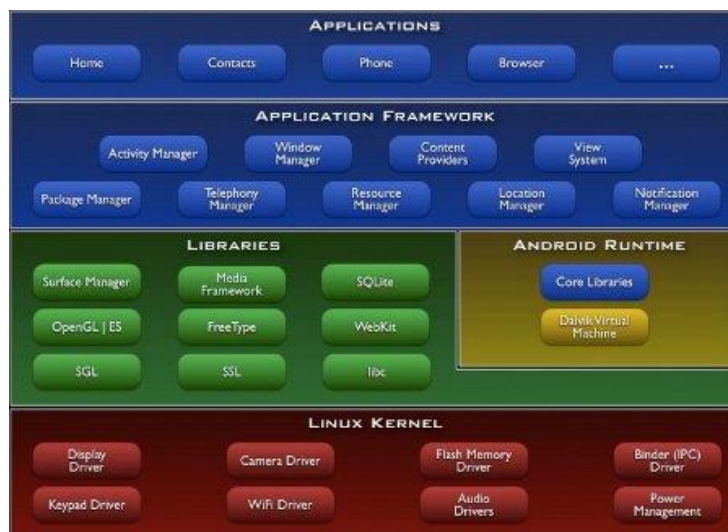


Figura 27: Camadas da Arquitetura Android ⁹

A primeira camada é responsável por disponibilizar um conjunto de aplicações base, disponibilizadas de origem pelo sistema operativo, tal como a agenda de contactos, a caixa de mensagens, o browser para aceder a *web*, entre outras aplicações.

A segunda camada corresponde à *framework* de aplicações. Esta camada oferece o acesso a funções específicas do equipamento onde corre o sistema operativo Android (gestão de ficheiros, localização, telefonia, entre outros).

A terceira camada é responsável por guardar as bibliotecas que são disponibilizadas ao programador para o desenvolvimento de aplicações Android. Para além disso, é nesta camada que se encontra a máquina virtual Java para Android (*Dalvik Virtual Machine*).

A última camada corresponde ao Kernel Linux da plataforma. Esta camada é responsável por tratar da gestão de memória, processos nucleares à plataforma, segurança e gestão do hardware.

4.2.2 Funcionamento offline

Um aspeto muito importante de referir é que a aplicação móvel não necessita de estar ligada à internet para a utilização da maioria das funcionalidades da mesma, ou seja, a aplicação funciona quase sempre em modo *offline*. Apenas é necessária a utilização da rede móvel para as seguintes duas operações:

- Sincronização dos dados: Para atualizar os conteúdos presentes na aplicação móvel, é necessária a utilização da rede móvel para efetuar a transferência de conteúdos entre a aplicação móvel e o servidor. Também é utilizada a rede móvel

⁹ <http://www.android-app-market.com/android-architecture.html>

no sentido contrário, para enviar os dados estatísticos da aplicação móvel para o servidor;

- Partilha de dados nas redes sociais: Para efetuar a partilha de dados nas redes sociais Twitter e Facebook é necessária a conexão à internet.

Todas as restantes funcionalidades da aplicação não necessitam a utilização da internet para o bom funcionamento das mesmas.

4.2.3 Manutenção

A componente mobile foi desenvolvida em Java uma vez que é a linguagem usada pelo sistema operativo Android. Quanto ao *back-end*, foi utilizada a linguagem Python uma vez que a *framework* Django, utilizada para a criação do *back-end*, é uma *framework* Python. Foi escolhida a *framework* Django de modo a manter a compatibilidade com os restantes projetos da empresa.

O código de *back-end* está estruturado de acordo com a *design pattern Model View Controller (MVC)*, uma vez que encoraja os programadores a dividirem o seu código em blocos autocontidos, com objetivos bem definidos, desde a gestão de dados, passando pela lógica de negócio ou até mesmo código relacionado com a interface de interação com o utilizador.

4.2.4 Componente servidor

A componente servidor representa o *back-end* da aplicação e é formada por uma *API REST* que permite à aplicação móvel descarregar os conteúdos disponíveis e necessários. Existe ainda a base de dados do servidor que terá de ser sincronizada com a do dispositivo móvel. A sincronização realiza-se sempre que o dispositivo móvel se liga à internet, efetuando atualização dos conteúdos.

Para a criação da API REST, é utilizada a *framework Django* [29]. Esta é uma *framework open-source* de desenvolvimento para a Web, escrita em *Python*, que segue a *design pattern MVC*. Esta *framework* oferece algumas ferramentas que melhoram a experiência do utilizador, nomeadamente uma interface de administração gerada automaticamente, bem como um *Object-relational mapping (ORM)* flexível, oferecendo suporte para vários tipos de bases de dados. Para criar a API REST, foi utilizado o projeto *Tastypie* [30], tratando-se de um *plugin* para *Django*. Este projeto apresenta as seguintes características:

- Permite gerar respostas em formato *Javascript Object Notation (JSON)*, *Yet Another Markup Language (YAML)* e *Extensible Markup Language (XML)*;
- Relaciona-se diretamente com os mecanismos internos do Django.

Mais detalhes sobre esta biblioteca podem ser encontrados na secção 2.8 presente no Estudo do Estado da Arte.

O formato de dados utilizado para a representação e troca de dados é o formato JSON. Mais uma vez, a escolha de utilização do formato JSON permite manter compatibilidade com os restantes projetos da empresa. Outra razão para o uso de JSON é o

facto de tratar-se de um formato bem mais leve do que XML [63]. Uma vez que será usada a linguagem Java do lado da aplicação móvel, torna-se bem mais fácil o uso de JSON para fazer *parsing* às mensagens. Na secção 4.5 podem ser encontrados detalhes sobre os métodos que compõem a API REST.

No que toca ao armazenamento de dados, o servidor faz o uso de uma base de dados POSTGRESQL para armazenamento de conteúdos. Uma das razões para a utilização deste motor de base de dados é que algumas das *API's* desenvolvidas pela empresa usam este motor, como por exemplo a aplicação *mesh-t*, uma aplicação dirigida para o turismo. Desta forma, consegue-se manter compatibilidade com as *API's* desenvolvidas pela empresa. Mais motivos para a utilização do motor PostgreSQL podem ser encontrados na secção 2.9 do capítulo do Estudo do Estado da Arte.

4.2.5 Componente Mobile

Como já referido anteriormente, existe uma componente Mobile que corresponde a uma aplicação móvel para o sistema operativo Android. A aplicação móvel desenvolvida é capaz de comunicar com o servidor (API REST) para efetuar a sincronização de dados entre a base de dados do servidor e a base de dados da aplicação móvel. Como já foi referido na secção 4.2.2 deste capítulo, a sincronização dos dados é uma das componentes que requer ligação à internet. A sincronização é efetuada sempre que o dispositivo tem ligação à internet.

4.2.6 Componente de visualização de estatísticas

Esta componente foi desenvolvida recorrendo a um *browser* para a representação dos dados. Após alguma pesquisa e comparação de ferramentas para representação de dados estatísticos em mapas, chegou-se à conclusão que a melhor ferramenta a usar é a biblioteca *PolyMaps* [17]. Trata-se de uma biblioteca Javascript para representação e criação dinâmica de mapas em *web-browsers* modernos. PolyMaps suporta uma variedade de representações visuais para dados vetoriais, para além da cartografia usual do OpenStreetMap, Bing, entre outros provedores de mapas. A biblioteca usa *Scalable Vector Graphics (SVG)* para exibir informações, permitindo usar regras CSS para definir o *design* dos dados. A tabela 6 presente na secção 2.4 do Estado da Arte resume as principais diferenças entre as ferramentas aqui apresentadas. Uma vez que a API REST foi criada através da *framework* Django, esta também foi utilizada para a criação da componente de visualização de estatísticas, integrando a biblioteca javascript PolyMaps no projeto Django. Os dados apresentados são recolhidos pela aplicação móvel do Jardim Botânico e enviados para o servidor. A aplicação de visualização de estatísticas trata de buscar os dados guardados no servidor e proceder ao tratamento dos mesmos, para no final apresentar os resultados.

Métricas recolhidas

Apresentam-se aqui as métricas que são apresentadas na aplicação, através da recolha de informações pela aplicação móvel.

- Métrica 1: Recolher a localização do utilizador de 10 em 10 segundos ou 10 em 10 metros;

- Métrica 2: Verificar quais os itens mais/menos pesquisados/visualizados na aplicação;
- Métrica 3: Verificar quais os itens que mais “likes” recolhem;
- Métrica 4: Verificar quais os itens que são mais/menos partilhados nas redes sociais;
- Métrica 5: Verificar quais os itens que reúnem mais comentários nos *posts* associados às plantas;
- Permitir filtros temporais: É possível visualizar as métricas recolhidas num determinado espaço de tempo, escolhido pelo utilizador. O utilizador pode escolher visualizar os dados num intervalo de tempo definido por dias (por exemplo entre o dia 3 de junho e o dia 12 de junho), visualizar os dados num intervalo de tempo definido por horas (por exemplo entre as 9 de manhã e as 2 da tarde), ou então visualizar os dados num intervalo de tempo definido por dias e horas.

Para a métrica 1, a posição do utilizador é guardada numa tabela na base de dados, criada para esse efeito. Utilizando os dados guardados de localização, basta apenas ver quais as zonas mais/menos frequentadas.

Para as métricas 2,3,4 e 5, também são guardados os dados em tabelas criadas para o efeito, consoante a métrica a tratar. Os dados são tratados sob a forma de contagem. Consoante a métrica a analisar, é atribuído um valor de contagem a cada uma das plantas.

Para a representação das métricas identificadas acima, é aqui necessário fazer a distinção entre dados quantitativos e dados qualitativos. É aplicada uma representação diferente aos dados, consoante o seu tipo. Os capítulos 2.5.1 e 2.5.2 dão exemplos destas duas diferentes representações de dados (qualitativos e quantitativos).

Autenticação e segurança

O acesso à componente de visualização de estatísticas é limitado apenas a pessoas autorizadas para esse efeito. Para isso, foi utilizado o mecanismo de autenticação da *framework* Django. A *framework* Django disponibiliza um sistema de autenticação de utilizadores, que lida com contas de utilizadores, grupos, permissões, e sessões baseadas em *cookies*. O sistema de autenticação verifica se o utilizador é quem diz ser, assim como as permissões dele caso a autenticação esteja correta [59].

4.2.7 Sincronização dos dados

Uma componente muito importante para a arquitetura e para a aplicação móvel é a sincronização dos dados entre a aplicação e o servidor. A aplicação móvel tem como requisito funcionar desligada de qualquer rede internet, salvo aquando a partilha de dados nas redes sociais. A utilização da componente de redes sociais envolve obrigatoriamente a utilização de internet, para ser possível partilhar dados nas mesmas. A sincronização de dados entre a base de dados do servidor e a aplicação móvel, ocorre sempre que o utilizador se liga à internet. Recorrendo a mecanismos do Android, a aplicação é capaz de detetar os momentos em que ligações internet são iniciadas e proceder à sincronização dos dados.

Foram definidos e implementados mecanismos para a sincronização dos dados. Apresentam-se esses mecanismos para a sincronização dos dados entre a aplicação móvel e o servidor.

Funcionamento geral

Um dos mecanismos identificado e implementado para a sincronização dos dados consiste no seguinte fluxo de atividades:

1. Guardar do lado do servidor uma *Tag* que representa a versão dos dados que o servidor possui.
2. Guardar do lado do *smartphone* a mesma *Tag* de modo a verificar a consistência dos dados entre o servidor e a aplicação.
3. Quando a aplicação é iniciada e caso existe ligação à internet, a aplicação envia uma mensagem para o servidor para este enviar a *Tag*. Para isso, a aplicação invoca um método da API, que vai pedir ao servidor o valor da *Tag* presente no mesmo.
4. O servidor envia a *tag* para a aplicação móvel e são tomadas as seguintes decisões:
 - a. Caso o valor da *Tag*, do lado do servidor, seja igual ao valor da *Tag* presente do lado do *smartphone*, significa que os dados entre o servidor e a aplicação estão sincronizados. O servidor não precisa enviar dados para a aplicação.
 - b. Caso o valor da *Tag* do lado do servidor não tenha o mesmo valor da *Tag* presente no *smartphone*, significa que os conteúdos do lado do *smartphone* não estão sincronizados. Portanto o servidor vai ter que enviar os dados atualizados para a aplicação.
 - c. Após a sincronização, a *Tag* do lado do *smartphone* é atualizada para a versão mais recente.

A Figura 28 mostra um diagrama de sequência do processo descrito acima:

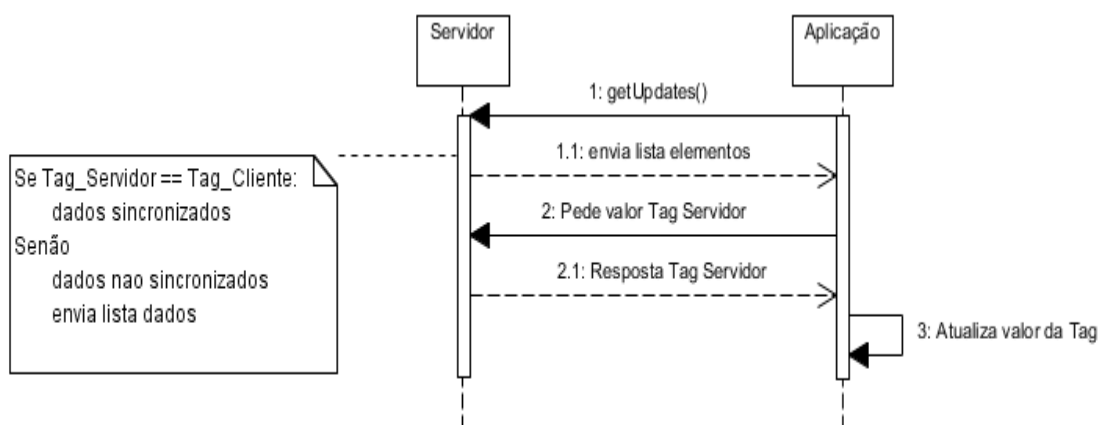


Figura 28: Diagrama de sequência para a sincronização dos dados

Caso Particular: Transferência de imagens

O mecanismo apresentado na secção anterior funciona bem para a transferência de textos, uma vez que se trata de conteúdos de pequena dimensão. No entanto, para envio de imagens entre o servidor e a aplicação, a solução apresentada acima não é muito adequada. O Envio de todas as imagens, do servidor para a aplicação, pode demorar muito tempo, tratando-se de conteúdos de grande dimensão, comparando com os textos. Portanto, para o envio de imagens foi identificado e implementado o seguinte mecanismo:

1. A aplicação envia uma mensagem para o servidor chamando o método que devolve a *Tag* presente do lado do mesmo;
2. Caso o valor da *Tag* enviada pelo servidor seja diferente do valor da *Tag* presente do lado da aplicação, o servidor vai enviar para a aplicação uma lista das imagens presentes no servidor, através da chamada do método da API que devolve à aplicação móvel a lista das imagens presentes;
3. Ao receber a lista, a aplicação vai comparar os nomes dos conteúdos nela presente com os nomes dos conteúdos presentes na lista de imagens da aplicação. Caso exista um nome de uma imagem na lista do servidor que não esteja na lista presente do lado da aplicação, esta vai enviar uma mensagem para o servidor com o nome da imagem a ser enviada pelo servidor. O servidor procede ao envio da imagem. No caso de existir uma imagem na lista da aplicação, que não existe na lista recebida do servidor, significa que a imagem com esse nome já não existe do lado do servidor. A aplicação procede a eliminação dessa imagem.

Com este mecanismo, apenas são enviadas para a aplicação as novas imagens que foram adicionadas do lado do servidor, minimizando o *overhead* em termos de tempo e de tráfego, comparando com o caso de enviar sempre todas as imagens para a aplicação. A Figura 29 mostra um diagrama de sequência do processo descrito acima.

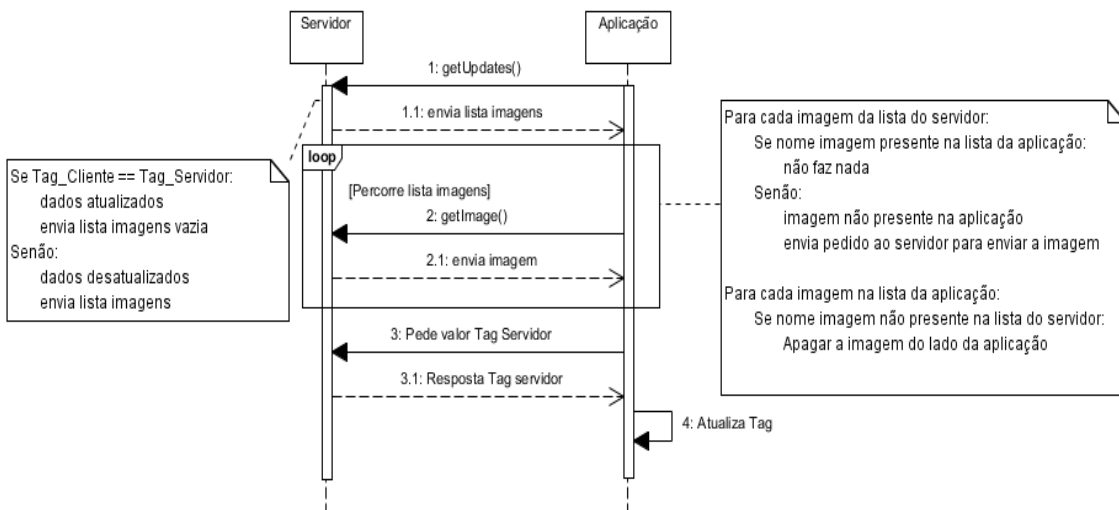


Figura 29: Diagrama de sequência para a sincronização das imagens

Transferência das coordenadas do utilizador guardadas no dispositivo móvel.

A aplicação é capaz de recolher as coordenadas do utilizador, em intervalos de tempo. Dado que a aplicação funciona desligada da rede, as coordenadas são guardadas no dispositivo móvel até o utilizador se ligar à rede. Uma vez ligado à rede, as coordenadas são enviadas para o servidor. De modo a não enviar dados replicados ou perderem-se dados com falhas de rede, é necessário garantir um mecanismo para a sincronização destes dados. Assim, foi identificado e implementado o seguinte mecanismo:

1. Quando o utilizador se liga à rede, a aplicação verifica se contém coordenadas para enviar para o servidor. Caso a aplicação contenha dados para enviar, o envio será efetuado da seguinte maneira:
 - a. São enviadas todas as coordenadas de uma vez, enviando um *array* de coordenadas em JSON. Também é enviada a data em que a coordenada foi obtida, podendo ser importante para tratamento de dados estatísticos.
 - b. Ao receber o *array* de coordenadas, o servidor devolve uma mensagem de resposta 200 OK de volta para a aplicação, confirmando que recebeu o *array* de coordenadas.
 - c. A aplicação presente no dispositivo móvel processa a resposta enviada pelo servidor e apaga as coordenadas que enviou para o servidor, caso receba uma mensagem de resposta 200 OK.

Gestão e representação da TAG

Para a representação da *Tag*, foi utilizado um inteiro (Por exemplo, considera-se inicialmente a *Tag* com o valor igual a 1000). Sempre que qualquer conteúdo é inserido, alterado ou apagado na base de dados, o valor da *Tag* é incrementado em uma unidade. Este mecanismo foi implementado recorrendo a um *Trigger* em cada uma das tabelas da base de dados, que dispara sempre que se verifica a inserção, alteração ou eliminação de conteúdos na base de dados. Ao disparar, o *Trigger* manda efetuar a atualização da *Tag*. Assim, é possível determinar quando existem alterações de conteúdos do lado do servidor.

4.3 Vistas da arquitetura

São apresentadas nesta secção duas vistas da arquitetura:

- A vista lógica, correspondente à lógica da arquitetura, mostrando as interações entre as várias componentes do sistema;
- A vista de desenvolvimento, identificando as tecnologias que foram usadas para o desenvolvimento de cada uma das componentes.

4.3.1 Vista lógica

A Figura 30 apresenta um diagrama da arquitetura do sistema, assim como as interações entre cada uma das componentes. O sistema consiste numa arquitetura Cliente-Servidor, onde o cliente acede através do seu *smartphone* à aplicação móvel, que comunica com o servidor para sincronização dos dados. A aplicação comunica com o servidor efetuando pedidos REST, trocando dados no formato JSON. Existe ainda um sistema para visualização de estatísticas, acedido através de um *browser*.

Analisando a componente servidor, destacam-se as seguintes componentes:

- Uma camada de lógica, onde corre um *application Server*, neste caso o Django;
- Uma camada de persistência, onde corre o motor de base de dados, neste caso o POSTGRESQL.

A Camada da lógica do servidor é a camada onde corre a *framework* Django, seguindo a *pattern Model-View-Controller*. No entanto, o modelo MVC da *framework* Django apresenta algumas diferenças relativamente ao tradicional modelo MVC. Uma vez que a camada *Controller* é tratada pela própria *framework*, e dado que muitas das operações em Django ocorrem nos modelos, *templates* e vistas, Django tem sido referenciada como uma *framework Model-Template-View (MTV)* [57]. Num *design-pattern* MTV:

- M significa "*Model*", a camada de acesso a dados. Esta camada contém toda a informação sobre os dados: como acedê-los, como validá-los, quais os comportamentos que têm e as relações entre os dados.
- T significa "*Template*", a camada de apresentação. Esta camada contém decisões relacionadas com a apresentação: como algo deve ser exibido numa página Web ou outro tipo de documento.
- V significa "*View*", camada de lógica do negócio. Esta camada contém a lógica que acede aos modelos. É visto como a ponte entre modelos e *templates*.

Como já referido anteriormente, o servidor consiste numa *API REST*. Esta API comunica com a camada de persistência (repositório de dados) responsável por guardar os dados da aplicação. A componente Mobile, comunica com o servidor, de modo a manter os seus dados sincronizados com os dados presentes na base de dados do servidor. Finalmente, a componente de visualização de estatísticas, que consiste numa aplicação web, comunica também com o servidor de modo a receber os dados que necessitam ser tratados para permitir a visualização de estatísticas.

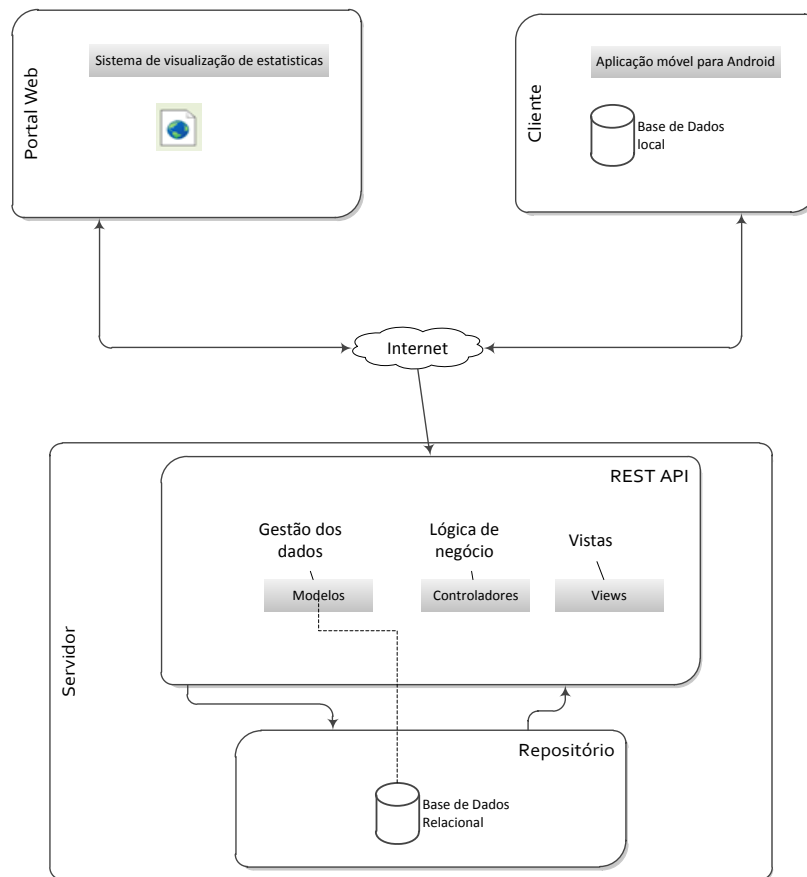


Figura 30: Vista lógica da arquitetura

4.3.2 Vista de desenvolvimento

Pode-se visualizar na Figura 31 um diagrama com as tecnologias utilizadas em cada uma das componentes que foram desenvolvidas. Como já referido anteriormente, a framework Django foi utilizada para a implementação do servidor, utilizando portanto a linguagem de programação Python. O motor de base de dados utilizado do lado do servidor foi o *POSTGRESQL*. No que diz respeito à aplicação móvel, foi utilizado o sistema operativo Android, recorrendo à linguagem de programação Java para o desenvolvimento. A base de dados utilizada na aplicação móvel é *SQLite*.

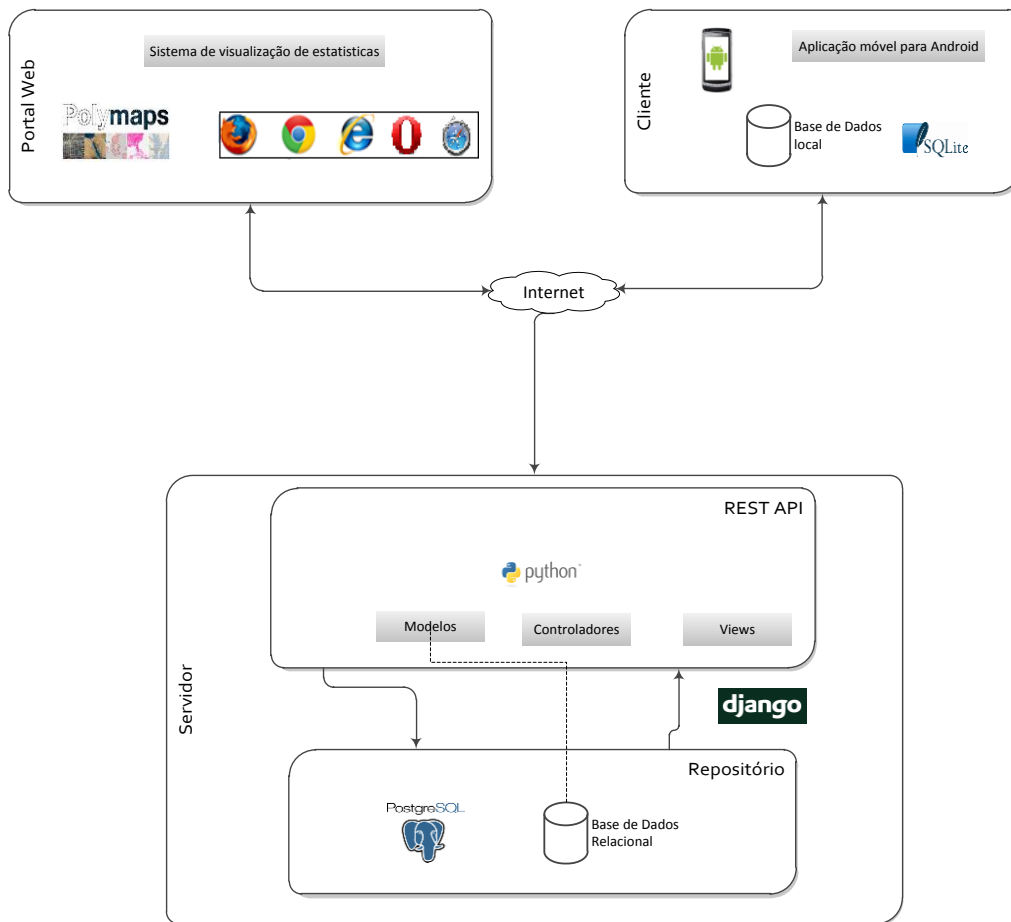


Figura 31: Vista de desenvolvimento da arquitetura

4.4 Modelo de dados

Nesta secção é apresentado o modelo de dados utilizado para guardar todos os conteúdos da aplicação. O modelo é apresentado primeiro sob a forma de um diagrama de classes (Figura 32) e depois sob a forma de um diagrama de entidades/relacionamentos (Figura 33). Uma descrição sobre a utilidade de cada tabela pode ser encontrada a seguir à Figura 33.

4.4.1 Diagrama de classes

Diagrama de classes

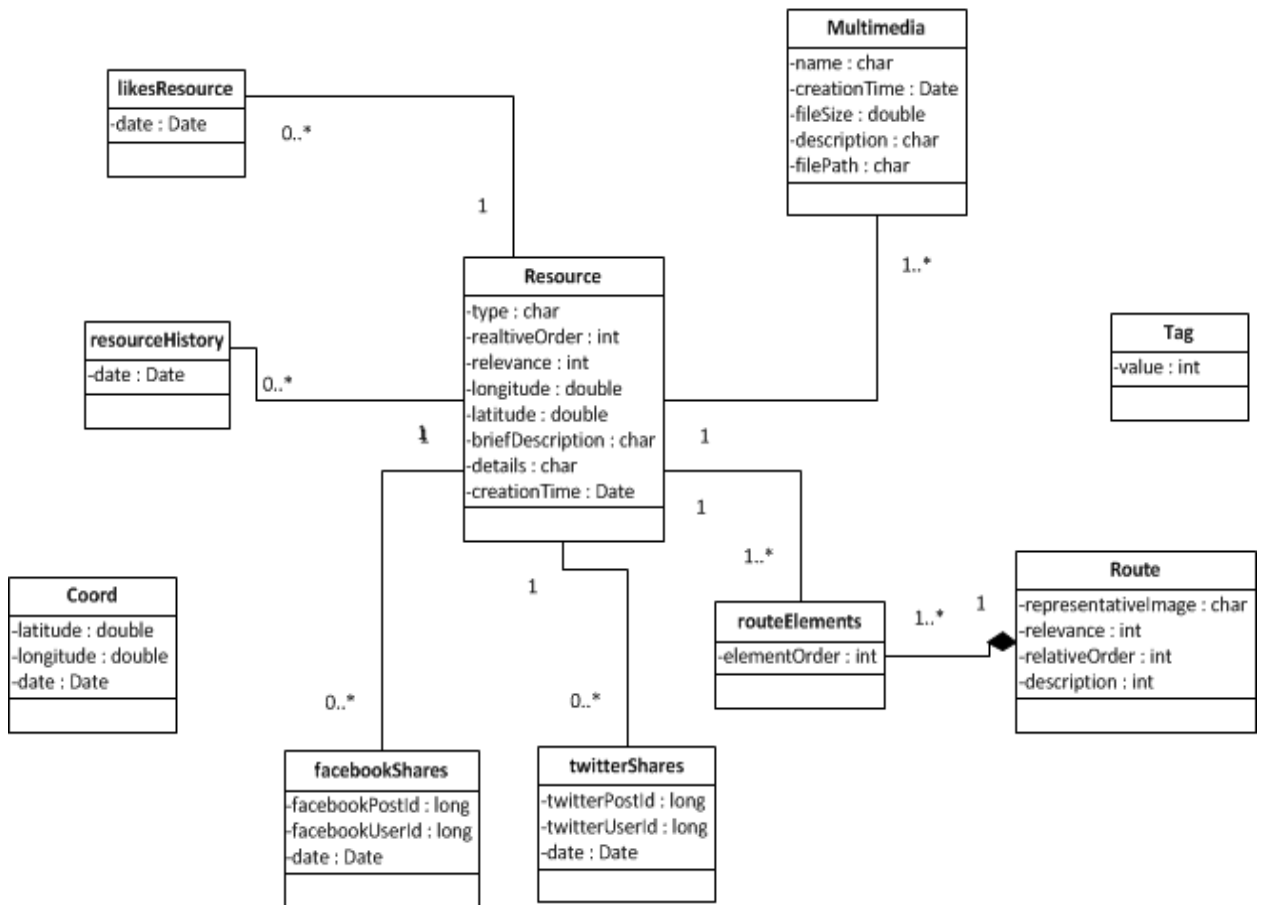


Figura 32: Representação do diagrama de classes

4.4.2 Diagrama de entidades/relacionamentos

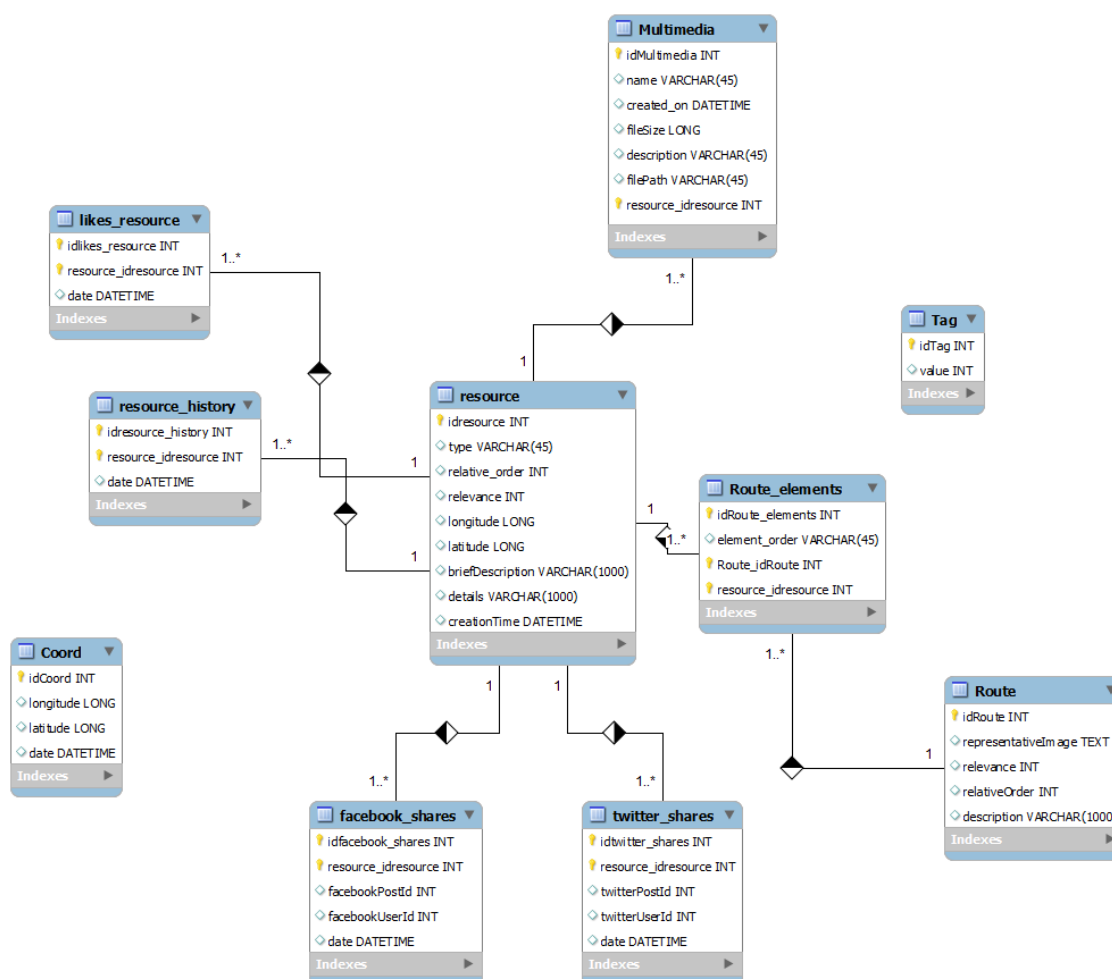


Figura 33: Representação do modelo de dados

Resource: Esta tabela é utilizada para guardar as plantas presentes no Jardim Botânico. Representa a parte central do modelo de dados, uma vez que muitas das restantes tabelas estão associadas a esta tabela. Permite guardar informação sobre a localização da planta (longitude, latitude), assim como a ordem e importância em que vai aparecer na aplicação. Também é guardada uma descrição da planta bem como detalhes sobre ela e a data de criação da mesma. Todas as plantas são um recurso.

Multimédia: representa qualquer recurso multimédia associado a um determinado recurso (plantas). Os recursos multimédia podem ser de tipo texto, imagem, vídeo.

Route: representa as rotas pré-definidas existentes no Jardim. Guarda uma descrição da rota assim como a ordem e importância em que vai aparecer na aplicação.

Route_elements: representa cada um dos troços associados a uma determinada rota. Existe o atributo “element_order” que permite definir a ordem dos vários troços dentro de cada rota.

resources_history: representa o modelo de dados que guarda o histórico dos recursos (plantas) visualizados pelos utilizadores. Esta tabela vai ser útil para posteriormente efetuar análises estatísticas sobre os conteúdos mais visualizados.

Coord: representa o modelo de dados que trata de guardar as coordenadas dos utilizadores de x em x tempo. Esta tabela é utilizada para efetuar análises estatísticas sobre os locais do Jardim onde os visitantes passam mais tempo.

Facebook_shares, twitter_shares Estas duas tabelas são utilizadas para guardar as partilhas de *posts*, por parte dos utilizadores, nas duas principais redes sociais. Utilizando estas tabelas, é possível efetuar análises estatísticas sobre os conteúdos mais partilhados nas redes sociais. Para aceder aos dados dos utilizadores nas redes sociais, é utilizado um identificador, único para cada utilizador, que é obtido sempre que é realizada uma operação numa das redes sociais. Sempre que um utilizador realiza uma operação de partilha, é extraído da resposta da operação, em JSON, o identificador do utilizador para posteriormente poder ser utilizado para obter os dados sobre as suas partilhas. As variáveis *facebook_user_id*, *twitter_user_id* permitem guardar o identificador do utilizador, consoante a rede social utilizada. De maneira análoga, é guardado o id dos *posts* efetuados pelos utilizadores nas redes sociais, de modo a poder aceder posteriormente a esses *posts*. As variáveis *facebook_post_id* e *twitter_post_id* são utilizadas para guardar o id dos *posts* efetuados, consoante a rede social utilizada.

Tag: Esta tabela é usada para representar o valor da *Tag* presente do lado do servidor. O atributo *value* representa a versão dos dados presentes do lado do servidor.

4.5 API REST

Para permitir o acesso aos dados armazenados no servidor, a empresa tem definida uma API REST, utilizada em vários projetos da empresa. Para a definição e implementação da API REST da aplicação desenvolvida, foi, numa primeira fase, analisada a solução existente, afim de adaptá-la às necessidades deste projeto. A API serve de base à definição dos métodos utilizados pelos controladores da aplicação. Esta interface está desenvolvida num modelo MVC. As secções 4.5.1 e 4.5.2 apresentam uma listagem dos métodos que compõem a API REST. É feita a distinção entre os módulos já existentes na solução da empresa (secção 4.5.2), que necessitaram de adaptação em função das necessidades do projeto, e entre os métodos que foram implementados de raiz (secção 4.5.1).

Mais especificamente, os métodos relacionados com a recolha de dados para tratamento estatístico foram implementados de raiz e adicionados à API existente, utilizada pela empresa em grande número de projectões. São eles os métodos que permitem guardar as coordenadas dos utilizadores no interior do recinto assim como os métodos relacionados com *"likes"* de recursos ou partilhas em redes sociais. Os métodos relacionados com os recursos(plantas), conteúdos multimédias e rotas foram utilizados da API existente, não necessitando a sua criação de raiz. Apenas são utilizados os modelos desejados da API existente, assim como os atributos necessários dentro de cada modelo.

4.5.1 Métodos implementados de raiz:

Coordenadas: Este módulo foi implementado para permitir à API receber e disponibilizar as coordenadas registadas dos utilizadores.

1. **Coordinates/:** Este método permite as operações GET, POST e DELETE de modo a listar, inserir ou apagar coordenadas na base de dados. Também permitir filtrar as coordenadas em função da data. Ao efetuar uma operação GET são devolvidas todas as coordenadas presentes no servidor.

Social_Networks: Este módulo foi implementado para permitir à API receber, disponibilizar ou apagar dados sobre as partilhas de “*posts*”, realizados pelos utilizadores nas três principais redes sociais.

1. **Facebook_shares/:** Este método permite as operações GET, POST e DELETE de modo a listar, inserir ou apagar partilhas de *posts* efetuadas na rede social *Facebook*.
2. **Twitter_shares/:** Este método permite as operações GET, POST e DELETE de modo a listar, inserir ou apagar partilhas de *posts* efetuadas na rede social *Twitter*.

Resources_History: Este módulo foi implementado para permitir à API receber, disponibilizar ou apagar dados sobre o histórico dos recursos visualizados com a aplicação.

1. **Resources_history/:** Este método permite efetuar as operações GET, POST e DELETE de modo a listar, inserir ou apagar o histórico de plantas visualizadas na aplicação.

Resources_Likes: Este módulo foi implementado para permitir à API receber, disponibilizar ou apagar dados sobre os likes dos recursos (plantas) visualizados com a aplicação.

1. **Resources_likes/:** Este método permite efetuar as operações GET, POST e DELETE de modo a listar, inserir ou apagar os likes de plantas visualizadas na aplicação.

4.5.2 Métodos adaptados da API existente

Multimedia: Este módulo já se encontra implementado na API existente. É utilizado para permitir à API guardar, disponibilizar, apagar ou alterar conteúdos multimédia.

1. **Multimedia/:** Este método permite as operações GET e POST de modo a listar todos os conteúdos multimédia presentes na base de dados e enviar um novo objeto para a base de dados. No Anexo VI encontram-se mais detalhes sobre a chamada deste método, tais como o retorno do método e os parâmetros a enviar.
2. **Multimedia/{{id}}:** Este método permite as operações GET, PUT e DELETE de modo a listar, modificar ou apagar um determinado conteúdo multimédia em função do seu identificador na base de dados (id). No Anexo VI encontram-se mais detalhes sobre a chamada deste método, tais como o retorno do método e os parâmetros a enviar.

Resources: Este módulo já se encontra implementado na API existente. É utilizado para permitir à API guardar, disponibilizar, apagar ou alterar recursos, bem como informação associada a um recurso. Neste caso, considerámos que um recurso representa uma determinada planta.

1. **Resources/:** Este método permite as operações GET e POST de modo a listar todas as plantas presentes na base de dados e enviar um novo objeto para a base de dados. No Anexo VI encontram-se mais detalhes sobre a chamada deste método, tais como o retorno do método e os parâmetros a enviar.
2. **Resources/{id}:** Este método permite as operações GET, PUT e DELETE de modo a listar, modificar ou apagar uma determinada planta em função do seu identificador na base de dados (id). No Anexo VI encontram-se mais detalhes sobre a chamada deste método, tais como o retorno do método e os parâmetros a enviar.

Routes: Este módulo já se encontra implementado na API existente. É utilizado para permitir à API guardar, disponibilizar, apagar ou alterar rotas pré-definidas, assim como determinados troços de uma rota ou informações sobre uma rota.

1. **Routes/:** Este método permite as operações GET e POST de modo a listar todas as rotas presentes na base de dados e enviar um novo objeto para a base de dados. No Anexo VI encontram-se mais detalhes sobre a chamada deste método, tais como o retorno do método e os parâmetros a enviar.
2. **Routes/{id}:** Este método permite as operações GET, PUT e DELETE de modo a listar, modificar ou apagar uma determinada rota em função do seu identificador na base de dados (id). No Anexo VI encontram-se mais detalhes sobre a chamada deste método, tais como o retorno do método e os parâmetros a enviar.
3. **Route_elements/:** Este método permite as operações GET, POST, PUT e DELETE de modo a listar, inserir, modificar ou apagar um elemento de uma rota. No Anexo VI encontram-se mais detalhes sobre a chamada deste método, tais como o retorno do método e os parâmetros a enviar.

4.5.3 Autenticação

A API existente disponibiliza um mecanismo para limitar o acesso aos dados, protegendo contra acessos a pessoas não autorizadas. O mecanismo consiste num sistema de *login*, permitindo apenas o acesso aos dados a pessoas que estejam autorizadas. Um utilizador deve ter um *username* e password para poder entrar, assim como um *access token* gerado quando é registado um novo utilizador. Assim, qualquer pedido de acesso a dados requer o envio das credenciais do utilizador.

aumentada, uma análise das API's existentes para integração com as redes sociais e uma análise de sistemas de gestão de dados. No final desta atividade foi entregue um relatório com o estado da arte.

Deliverables: Versão 0.5 do Estado da Arte (31 Dezembro de 2012); Versão 1.0 do Estado da Arte (20 Janeiro de 2013).

Atividade: Análise e especificação dos requisitos do sistema

Datas: De 14 de Outubro até 11 de Novembro

Descrição: Esta atividade consiste na identificação e documentação dos requisitos a implementar na aplicação. Para isso, foi primeiro elaborado um conjunto de *user-stories*. Numa segunda fase e com base nas *user-stories*, foi feita a elaboração de casos de uso para descrever os requisitos funcionais e não-funcionais, com maior grau de detalhe. No final, foi entregue um relatório de especificação de requisitos.

Deliverables: Relatório de especificação de requisitos.

Atividade: Definição da arquitetura da solução

Datas: De 21 de Outubro até 11 de Novembro

Descrição: Esta atividade consiste na definição da arquitetura do sistema. Esta parte está dividida na realização das seguintes subactividades: Análise da arquitetura de outras soluções da empresa; A elaboração dos mecanismos de sincronização de dados entre o servidor e a aplicação móvel; A elaboração das vistas da arquitetura. Criou-se um diagrama para a vista lógica da arquitetura e para a vista de desenvolvimento; A elaboração do modelo de dados; No final da atividade foi entregue um relatório da arquitetura do sistema.

Deliverables: Relatório da arquitetura do sistema

Atividade: Início do desenvolvimento

Data: De 12 de Novembro até de Fevereiro

Descrição: Esta atividade tem como objetivo o início do desenvolvimento da aplicação, começando a desenvolver a componente servidor. Para isso, foi primeiro necessário adquirir conhecimentos adicionais para a criação do servidor, recorrendo a tutoriais e documentação.

5.1.2 Calendário previsto inicialmente para o segundo semestre

Identificação	Nome da tarefa	Fev 2013			Mar 2013				Abr 2013				Mai 2013				Jun 2013						
		10-2	17-2	24-2	3-3	10-3	17-3	24-3	31-3	7-4	14-4	21-4	28-4	5-5	12-5	19-5	26-5	2-6	9-6	16-6	23-6		
1	Criação da componente servidor	■																					
2	Criação dos módulos base da aplicação	■																					
3	Criação do módulo de realidade aumentada																						
4	Criação do sistema de visualização de estatísticas																						
5	Testes																						
6	Fim do estágio																						

Descrição das atividades

Atividade: Criação da componente servidor

Datas: De 11 de Novembro até 25 de Fevereiro

Descrição: Esta atividade consiste no desenvolvimento da componente servidor.

Atividade: Criação dos módulos base da aplicação

Datas: De 26 de Fevereiro até 29 de Março

Descrição: Esta atividade tem como objetivo a criação dos módulos base da aplicação. Destacam-se os seguintes módulos: Visualização de informação, módulo que permite visualizar informações de uma determinada planta ou rota; pesquisa de dados, módulo que permite pesquisar por plantas ou rotas; Posicionamento e navegação, módulo que permite ao utilizador visualizar a sua posição no interior do Jardim; redes sociais, módulo que permite ao utilizador partilhar dados nas redes sociais.

Atividade: Criação do módulo de realidade aumentada

Datas: De 1 de Abril até 15 de Abril

Descrição: Esta atividade tem como objetivo a criação do módulo de realidade aumentada na aplicação, permitindo georreferenciação de dados. Com isto, é possível ao utilizador visualizar a localização das plantas no Jardim através da realidade aumentada. Também é possível seguir uma rota através da realidade aumentada

Atividade: Criação do sistema de visualização de estatísticas

Datas: De 16 de Abril até 10 de Maio

Descrição: Esta atividade tem como objetivo a criação de um sistema de visualização de estatísticas através de um *browser*, apresentando estatísticas sobre dados recolhidos a partir da aplicação móvel.

Atividade: Testes

Datas: De 13 de Maio até 23 de Junho

Descrição: Esta atividade consiste na execução de testes de aceitação da aplicação, verificando se os requisitos da aplicação estão corretamente implementados e resolvendo os *bugs* que possam aparecer. No final desta atividade deverá ser entregue o relatório de execução de testes, um manual de instalação/configuração, um manual de administração e um manual de utilização.

Deliverables: Relatório de execução de testes; Manual de instalação/configuração; manual de administração; manual de utilização, manual de conceção/desenvolvimento.

5.1.3 Desvios na calendarização do segundo semestre em relação ao que tinha sido definido inicialmente

O planeamento para o segundo semestre teve que ser redefinido uma vez que o autor participou no desenvolvimento de uma outra aplicação móvel durante a primeira parte do segundo semestre. Foi feito um planeamento que permitisse conciliar o desenvolvimento da aplicação móvel para o Jardim Botânico com o desenvolvimento da outra aplicação móvel (mais detalhes sobre a mesma podem ser encontrados no Capítulo 6). Assim, é apresentado a seguir o diagrama de Gantt final relativo ao segundo semestre.

Identificação	Nome da tarefa	Fev 2013			Mar 2013				Abr 2013				Mai 2013			Jun 2013					
		10-2	17-2	24-2	3-3	10-3	17-3	24-3	31-3	7-4	14-4	21-4	28-4	5-5	12-5	19-5	26-5	2-6	9-6	16-6	23-6
1	Criação da componente servidor	█																			
2	Desenvolvimento de uma outra aplicação móvel		█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
3	Escrita de um artigo para uma revista científica							█	█	█	█	█	█	█	█	█	█	█	█	█	█
4	Criação dos módulos base da aplicação						█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
5	Criação do módulo de realidade aumentada								█	█	█	█	█	█	█	█	█	█	█	█	█
6	Criação do sistema de visualização de estatísticas															█	█	█	█	█	█
7	Testes e correções									█	█	█	█	█	█	█	█	█	█	█	█
8	Fim do estágio																				█

Comparando com o diagrama do planeamento previsto para o segundo semestre, foram adicionadas duas tarefas. A primeira para o desenvolvimento da outra aplicação móvel, com uma duração de 1 mês e meio. A segunda para a escrita de um artigo para tentativa de submissão numa revista científica, com duração da tarefa de quase três semanas. Salvo a tarefa do desenvolvimento do servidor, todas as restantes tarefas tiveram que ser adiadas em relação ao planeamento inicial. Para além disso, foi alterada a tarefa de “Testes e correções” e alocada durante todo o desenvolvimento da aplicação móvel e da aplicação de visualização de estatísticas. À medida que funcionalidades iam sendo implementadas, estas iam também sendo testadas, procedendo-se à correção de falhas encontradas. A criação do módulo de realidade aumentada teve que iniciar durante a tarefa de criação dos módulos base da aplicação para o Jardim Botânico, de modo a conseguir-se obter resultados suficientes para serem publicados no artigo a submeter. Apresentam-se a seguir as atividades adicionadas em relação ao planeamento inicial.

Descrição das atividades adicionadas

Atividade: Desenvolvimento de uma outra aplicação móvel orientada ao turismo

Datas: De 26 de Fevereiro até 5 de Abril

Descrição: Esta atividade consiste no desenvolvimento de uma outra aplicação móvel dirigida para o turismo. Mais detalhes sobre a aplicação podem ser encontrados na secção 6.6, no capítulo do desenvolvimento.

Atividade: Escrita de um artigo para tentativa de submissão numa revista científica.

Datas: De 27 de Março até 19 de Abril

Descrição: Esta atividade consiste na escrita de um artigo para tentativa de submissão na revista científica RISTI (Revista Ibérica de Sistemas e Tecnologias de Informação)

5.2 Metodologia e artefactos utilizados no desenvolvimento

Apresenta-se nesta secção a metodologia de desenvolvimento seguida durante o desenvolvimento do projeto. São apresentados também alguns artefactos que complementaram o desenvolvimento do projeto.

5.2.1 Metodologia de desenvolvimento.

Durante o desenvolvimento do projeto, não foi seguida “à risca” nenhuma metodologia em concreto. A empresa costuma seguir a metodologia ágil *Scrum* na maioria dos projetos no entanto, devido à natureza deste projeto, não se pode considerar que esta tenha sido seguida. Uma das características que diferencia este projeto é que a equipa de desenvolvimento é apenas constituída pelo estagiário. Apenas na fase de prototipagem da aplicação participou mais um elemento, um *designer* da empresa, ajudando a definir uma interface de qualidade para a aplicação móvel.

Foi adotada uma abordagem iterativa durante o desenvolvimento do projeto. Cada iteração tem uma duração de uma semana. Ao fim de cada iteração é realizada uma reunião entre o estagiário e o gestor de projeto de modo a verificar o trabalho desenvolvido durante a iteração, o trabalho que ficou por fazer assim como realizar o planeamento do trabalho a efetuar na semana seguinte.

5.2.2 Artefactos utilizados durante o desenvolvimento

Redmine

Para a gestão dos projetos, a empresa tem como boa prática usar a ferramenta Redmine. Trata-se de uma aplicação *web*, escrita em *ruby-on-rails* que permite gerir os projetos. Esta aplicação permite criar e registar as tarefas de cada um dos desenvolvedores como também registar o tempo que foi necessário ao desenvolvimento de cada uma das tarefas. Para além disso, também é possível submeter documentos mas também usar a Wiki para cada projeto.

Tal como para os outros projetos da empresa, o desenvolvimento da Aplicação, assim como as fases anteriores(definição da arquitetura, definição dos requisitos, estudo do estado da arte, prototipagem, etc..) foram geridos através do Redmine. Todas as tarefas a serem desenvolvidas pelo autor eram criadas na plataforma Redmine. À medida que as tarefas eram concretizadas, era registado o estado das mesmas assim como o tempo gasto na realização de cada uma das tarefas. Também era registado o tempo gasto em reuniões. A figura 39 ilustra a utilização do Redmine para a listagem das tarefas.

Tracker	Status	Priority	Subject	Assignee	Category	Target version	Start date	Due date	% Done
Task	New	Normal	Primeira iteração de testes no recinto + preparação dos testes	Flávio Pereira			2013-06-05		100%
Task	New	Normal	[Polymaps] - Criação da estrutura do site/manipulação da biblioteca para criação do mapa	Flávio Pereira			2013-06-11		0%
Task	Assigned	Normal	[Polymaps] - Limitar o mapa	Flávio Pereira			2013-06-12	2013-06-17	100%
Task	Assigned	Normal	[Polymaps] - Pop-ups nos pontos do mapa	Flávio Pereira			2013-06-12	2013-06-17	100%
Task	Assigned	Normal	[Polymaps] - Criar links na listagem das plantas	Flávio Pereira			2013-06-12	2013-06-17	100%
Task	Assigned	Normal	[Polymaps] - Criar filtro de horas	Flávio Pereira			2013-06-12		100%
Task	Assigned	Normal	[Polymaps] - Criar filtro de data	Flávio Pereira			2013-06-12	2013-06-17	100%
Bug	New	Normal	Broadcast Receiver para envio dos dados estatísticos	Flávio Pereira			2013-06-04	2013-06-04	100%
Task	New	Normal	Correções no servidor	Flávio Pereira			2013-05-31		100%
Bug	New	Normal	Carregamento de imagens	Flávio Pereira			2013-05-30		100%
Task	New	Normal	Redes Sociais - Guardar dados para estatísticas	Flávio Pereira			2013-05-29		100%
Bug	New	Normal	Pesquisas - Fazer corresponder a imagem à planta	Flávio Pereira			2013-05-29		100%
Bug	New	Normal	Realidade aumentada - Próximo ponto da rota mal definido	Flávio Pereira			2013-05-29		100%
Bug	New	Normal	Rotas - Visualização de pontos que não pertencem à rota	Flávio Pereira			2013-05-28		100%
Task	New	Normal	Rota - Inserir um nome na criação de uma nova rota	Flávio Pereira			2013-05-28		100%
Bug	New	Normal	Realidade aumentada - Deixar Tela sempre ativa	Flávio Pereira			2013-05-28		100%
Bug	New	Normal	Realidade aumentada - Imagem do POI muito pequena	Flávio Pereira			2013-05-28		100%
Task	Assigned	Normal	Realidade Aumentada - alterar o fundo dos POIs	Flávio Pereira			2013-05-28		100%
Bug	Assigned	Normal	Botão do lado direito	Flávio Pereira			2013-05-28		100%

Figura39: Tarefas criadas no Redmine

GIT

Para a gestão do código, foi utilizada a ferramenta Git. Trata-se de um sistema de controlo de versões de gestão de código fonte. Este sistema permite que cada programador tenha uma cópia local de todo o histórico de desenvolvimento, e que possa realizar *commits* mesmo estando offline. Para além disso, esta ferramenta facilita os processos de *branching e merging* (criação e fusão de ramos divergentes de código). Outra característica importante é que esta ferramenta está integrada com outras ferramentas utilizadas para a gestão do projeto como é o caso do Redmine. Esta ferramenta é utilizada para a gestão e para o controlo de versões de código fonte nos diversos projetos da empresa.

Capítulo 6

Desenvolvimento

O presente capítulo apresenta detalhes sobre a etapa de desenvolvimento das componentes que foram desenvolvidas. São apresentados detalhes sobre o código desenvolvido, tais como a estrutura do mesmo ou análise da responsabilidade dos ficheiros de código mais importantes. Também são apresentados os principais problemas encontrados durante a fase de desenvolvimento e respectivas soluções adotadas quando possível.

6.1 Desenvolvimento da componente servidor

Como já referido anteriormente, a componente servidor corresponde a uma API REST, desenvolvida através da *framework* Django e da biblioteca *Tastypie*. Um projeto Django segue por defeito a seguinte estrutura:

```
mysite/  
  manage.py  
  mysite/  
    __init__.py  
    settings.py  
    urls.py  
    wsgi.py
```

, onde o ficheiro `settings.py` é responsável por guardar as configurações do projeto, assim como endereço da base de dados, *templates* e aplicações usadas; o ficheiro `urls.py` é responsável por definir os `urls` criados para as diferentes vistas da aplicação; o ficheiro `manage.py` corresponde a uma linha de comandos que permite interagir com o projeto Django de várias maneiras; o ficheiro `wsgi.py` corresponde a um ponto de entrada para servidores *web* compatíveis com WSGI para servir o projeto. Os ficheiros apresentados tratam essencialmente das configurações de um projeto. Para criar uma aplicação, em Django, é necessário criar o projeto manualmente (dentro da estrutura apresentada), criando automaticamente uma pasta com a seguinte hierarquia de ficheiros:

```
projeto/  
  __init__.py  
  admin.py  
  models.py  
  tests.py  
  views.py
```

, onde o ficheiro `admin.py` é responsável por definir a interface de administração de dados da aplicação (interface própria do Django para gestão do *backend*); o ficheiro `models.py` permite criar os modelos de dados da aplicação. É neste ficheiro que são criados os

modelos definidos no modelo de dados apresentado na secção 4.4, na especificação da arquitetura; o ficheiro `tests.py` é responsável por definir os testes a serem efetuados para testar a aplicação; o ficheiro `views.py` é responsável por criar os “controladores” da aplicação que vão comunicar com os *templates* da mesma. Neste caso, tratando-se de uma API REST, o ficheiro `views.py` não é utilizado.

No desenvolvimento de uma API REST, para além dos ficheiros apresentados acima, é utilizado o ficheiro `resources.py`. Este ficheiro é responsável por definir os métodos a serem implementados pela API, através da utilização da biblioteca *tastypie*. Os métodos responsáveis pela gestão dos *resources* (plantas), multimédias e rotas já se encontravam implementados na API existente da empresa e utilizada em grande parte dos projetos da mesma e, portanto, disponíveis para serem consumidos pela aplicação móvel. Os restantes métodos, todos relacionados com o armazenamento de dados estatísticos, tiveram que ser criados em cima da API existente. A tabela 14 resume os métodos consumidos na API assim como a sua origem (existentes na API da empresa ou criados).

Método	Origem
Resource	Existente
Multimedia	Existente
Route	Existente
Route_Elements	Existente
FacebookShares	Novo
TwitterShares	Novo
GoogleShares	Novo
RouteLikes	Novo
MultimediaLikes	Novo
Coord	Novo
ResourceLikes	Novo
Tag	Novo

Tabela 14: Resumo dos métodos da API

O grande desafio presente no desenvolvimento do servidor foi pegar na API utilizada pela empresa, analisá-la e perceber quais os modelos e métodos que poderiam ser utilizados para a aplicação do Jardim Botânico. A API existente é formada por uma grande quantidade de tabelas e, perceber o modelo de dados da mesma não foi tarefa simples. Para além disso era também preciso testar os métodos que pareciam ser adequados para utilização. Uma vez determinados os métodos da API existente que iriam ser utilizados, foram adicionados os modelos e métodos necessários para o tratamento dos dados estatísticos.

6.2 Desenvolvimento da Aplicação móvel

Apresenta-se aqui as fases de prototipagem da aplicação móvel assim como a fase de desenvolvimento da mesma.

6.2.1 Prototipagem da aplicação móvel

A prototipagem da aplicação é importante para definir a interface da aplicação assim como as transições entre os vários ecrãs da mesma. Assim, numa primeira fase, foi feita uma prototipagem da aplicação por parte do autor, através da elaboração e de criação

de *mockups*, uma prática corrente para prototipagem de aplicações deste tipo. Após elaboração dos *mockups*, estes foram apresentados ao *designer* da empresa de modo a apresentar a ideia a desenvolver e posteriormente receber conselhos e ajustes no design da aplicação, de acordo com as indicações do designer, resultante da sua experiência na área. Assim, o Anexo II apresenta as soluções desenhadas pelo autor e a respectiva solução definida pelo designer para o mesmo ecrã. A Figura 34 apresenta uma dessas comparações para a utilização do mapa na aplicação móvel. Como é possível ver, existe uma grande diferença entre a solução desenhada pelo autor e a solução desenhada pelo *designer*. Esta fase de prototipagem e ajuda recebida por parte de um profissional de design foi importante para definir uma interface usável e de qualidade para a aplicação. A Figura 34 mostra a comparação entre um *mockup* criado pelo autor e um *mockup* criado pelo *designer*.

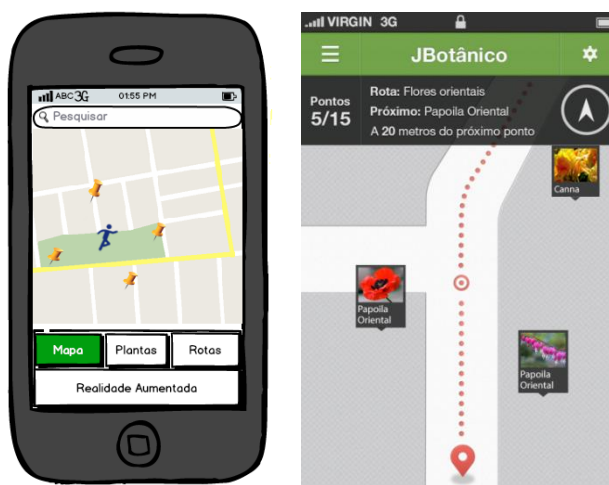


Figura 34: Comparação de um *mockup* criado pelo autor (figura da esquerda) com um *mockup* criado pelo *designer* (figura da direita)

6.2.2 Desenvolvimento da aplicação Android

São apresentados nesta secção os principais ficheiros que compõem o projeto da aplicação assim como a árvore de diretórios do mesmo. A apresentação desta estrutura possibilita desenvolvimentos futuros neste projeto de maneira mais simples. A Figura X apresenta a estrutura de diretórios da aplicação móvel.

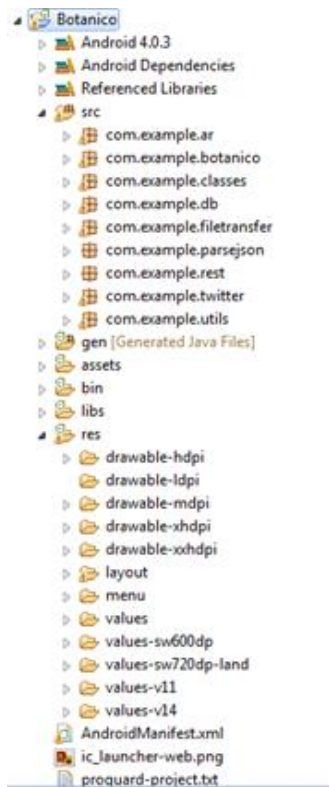


Figura 35: Estrutura do código de uma aplicação Android

Uma aplicação *Android* segue por defeito uma estrutura de diretórios, criada sempre que se cria uma nova aplicação. Apresentam-se os diretórios que compõem a aplicação *Android*:

Src: Diretório onde se encontram as classes Java, responsáveis pela criação das atividades e fragmentos da aplicação. Dentro desse diretório, encontram-se *packages*, cada uma contendo o código responsável por determinadas funcionalidades da aplicação. Esta divisão em *packages* permite uma melhor organização do código. Analisam-se a seguir as *packages* existente. Para cada uma delas, é apresentada uma tabela com as principais atividades inseridas e suas responsabilidades:

- **com.example.ar:** pasta que contém os ficheiros de código responsáveis pela componente de realidade aumentada.

Ficheiros	Descrição
AugmentedReality.java	Esta classe é responsável por criar o <i>layout</i> da parte da realidade aumentada, ou seja, inicializar os elementos gráficos a serem apresentados na câmara do dispositivo móvel.
DataSource.java	Classe que serve para a criação dos vários tipos de dados que podem ser representados pelo módulo de realidade aumentada.
SensorsActivity.java	Classe responsável por recolher os dados proveniente dos sensores do dispositivo tais como posição do utilizador, acelerómetro, entre outros.

Tabela 15: Principais ficheiros da package com.example.ar

- **com.example.botanico:** Esta pasta contém os principais ficheiros de código, responsáveis por implementarem a lógica da aplicação assim como as componentes gráficas de interface da aplicação móvel.

Ficheiros	Descrição
SplashActivity.java	Esta classe é responsável por apresentar o ecrã inicial da aplicação e, conseqüentemente, proceder às atualizações da aplicação.
MainActivity.java	Esta classe corresponde à classe principal da aplicação móvel. Implementa o <i>sliding menu</i> da direita e da esquerda assim como a barra de ações. É sobre esta classe que funcionam os vários fragmentos da aplicação.
LeftMenuFragment.java	Esta classe é responsável por criar e gerir o menu da esquerda da aplicação, contendo as principais ações de navegação da aplicação.
PlantasFragment.java	Esta classe é responsável pela apresentação de uma listagem completa das plantas presentes na aplicação.
MapaFragment.java	Esta classe é responsável por apresentar o mapa do Jardim Botânico, assim como as plantas na sua respectiva localização. Esta classe é utilizada tanto para o mapa geral como para as rotas e a procura de uma planta.
SwipeDetector.java	Esta classe é responsável por detectar movimentos específicos no ecrã do dispositivo móvel tais como deslocamentos com o dedo, etc..
RotasFragment.java	Esta classe é responsável por apresentar ao utilizador as rotas disponíveis na aplicação móvel.
BiPlantafragment.java	Esta classe é responsável por apresentar ao utilizador os detalhes de uma planta. Também permite procurar pela planta no mapa 2D, ou através da realidade aumentada. Permite a partilha de dados nas redes sociais.
NovaRotaFragment.java	Esta classe é responsável por permitir ao utilizador a criação da sua própria rota, escolhendo as plantas que deseja incluir na rota.

Tabela 16: Principais ficheiros da package com.example.botanico

- **com.example.classes:** pasta que contém os ficheiros de código responsáveis por criar os objetos presentes no modelo de dados da aplicação, tais como multimédia, *resources*, rotas, etc..

Ficheiros	Descrição
Coord.java	Esta classe é responsável por guardar objetos que representam coordenadas de um utilizador recolhidas pela aplicação, para efeitos estatísticos.
FacebookShares.java	Esta classe é responsável por criar objetos que representam partilhas de dados na rede social Facebook.
Multimedia.java	Esta classe é responsável por criar objetos que representam um conteúdo multimédia.

MultimediaLikes.java	Esta classe é responsável por criar objetos que representam um “like” de um determinado conteúdo multimédia.
Resource.java	Esta classe é responsável por criar objetos que representam um <i>resource</i> (uma planta).
ResourceHistory.java	Esta classe é responsável por criar objetos que representam históricos de visitas a uma determinada planta.
ResourceLikes.java	Esta classe é utilizada para criar objetos que representam um “like” a um <i>resource</i> (planta).
Route.java	Esta classe é responsável por criar objetos que representam uma rota.
RouteElements.java	Esta classe é responsável por criar objetos que representam elementos de uma rota.
RouteLikes.java	Esta classe é responsável por criar objetos que representam um “like” de uma determinada rota.
Tag.java	Esta classe é responsável por criar um objecto de tipo <i>Tag</i> , responsável por manter a versão dos dados presentes no servidor e na aplicação móvel.
TwitterShares.java	Esta classe permite a criação de objetos responsáveis por guardar uma partilha na rede social Twitter.

Tabela 17: Principais ficheiros da package com.example.classes

- **com.example.db:** pasta que mantém os ficheiros de código responsáveis por criar a base de dados local da aplicação móvel, assim como as operações de inserção, eliminação e listagem de dados da base na base de dados do dispositivo móvel.

Ficheiros	Descrição
DbDataSource.java	Esta classe é responsável por disponibilizar métodos para listar, inserir ou apagar dados presentes na base de dados do dispositivo móvel.
DatabaseHandler.java	Esta classe é responsável por criar as tabelas da base de dados da aplicação móvel.

Tabela 18: Principais ficheiros da package com.example.db

- **com.example.parsejson:** pasta responsável por manter os ficheiros de código utilizados para efetuar o *parsing* das diferentes respostas Json recebidas pelo servidor, contendo os objetos recebidos pelo servidor que vão passar para a base de dados local da aplicação móvel.

Ficheiros	Descrição
Location.java	Ficheiro responsável por fazer tratamento dos dados Json relacionados com dados de localização de objetos.
ObjectMultimedia.java	Ficheiro responsável por fazer tratamento dos dados Json relacionados com conteúdos multimédia.
ObjectResource.java	Ficheiro responsável por fazer tratamento dos dados Json relacionados com <i>resources</i> (plantas).
ObjectRoute.java	Ficheiro responsável por fazer tratamento dos

	dados Json relacionados com rotas.
ObjectTag.java	Ficheiro responsável por fazer tratamento dos dados Json relacionados com a Tag de controlo de versão de dados.
ResourceInfo.java	Ficheiro responsável por fazer tratamento dos dados Json relacionados com os detalhes de um resource (planta).
ResponseModel.java	Ficheiro responsável por fazer tratamento da estrutura global dos dados Json relacionados com um <i>resource</i> (planta).
ResponseMultimediaModel.java	Ficheiro responsável por fazer tratamento da estrutura global dos dados Json relacionados com conteúdos multimédia.
ResponseRouteModel.java	Ficheiro responsável por fazer tratamento da estrutura global dos dados Json relacionados com rotas.
ResponseTagModel.java	Ficheiro responsável por fazer tratamento da estrutura global dos dados Json relacionados com a Tag de consistência de dados do servidor.
RouteInfo.java	Ficheiro responsável por fazer tratamento dos dados Json relacionados com os detalhes de uma rota.

Tabela 19: principais ficheiros da package com.example.parsejson

- **com.example.rest:** pasta responsável por guardar os ficheiros de código utilizados para a criação da interface REST que efetua a comunicação entre a aplicação móvel e o servidor.

Ficheiros	Descrição
RequestMethod.java	Este ficheiro é uma enumeração dos métodos a serem utilizados na API REST.
RestClient.java	Este ficheiro define os métodos utilizados na troca de informação entre a aplicação móvel e a API REST.

Tabela 20: Principais ficheiros da package com.example.rest

- **com.example.twitter:** pasta responsável por manter os ficheiros de código responsáveis pela partilha de dados na rede social *Twitter*.

Ficheiros	Descrição
Twitt.java	Este ficheiro é responsável por efetuar uma partilha no Twitter do utilizador.
TwitterSession.java	Este ficheiro é responsável por criar uma sessão no Twitter, chamando a plataforma para o utilizador efetuar o <i>login</i> .

Tabela 21: Principais ficheiros da package com.example.twitter

Gen: Nesta pasta são gerados dados de maneira automática pelo compilador, para a compilação do projeto.

Res: Diretório onde se encontram os recursos utilizados pela aplicação, entre os quais:

- **Layout:** Diretório que contém os *layouts* da interface da aplicação.

- Menu: Diretório que contém os *layouts* da interface da aplicação.
- Values: Diretório que contém os estilos utilizados na aplicação tais como cores, tamanhos ou algumas *strings*.

Manifest.xml: Este ficheiro xml é necessário em qualquer aplicação Android, apresentando informações importantes sobre a aplicação para o sistema Android. Essas informações são necessárias para o sistema antes que este possa correr qualquer linha de código. Entre outras coisas, o *manifest* faz o seguinte [66]:

- Descreve as componentes da aplicação: as atividades, serviços, *broadcast receivers* e *contente providers*, que compõem a aplicação
- Declara as permissões que a aplicação deve ter para aceder a partes protegidas da API e para interagir com outras componentes da aplicação
- Declara o nível mínimo da API do Android que a aplicação requer.
- Lista as bibliotecas as quais a aplicação deve estar ligada.

O seguinte quadro exemplifica a definição das permissões no ficheiro manifest.xml de uma aplicação Android.

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.CAMERA" />
  <uses-permission
android:name="android.permission.ACCESS_WIFI_STATE" />
  <uses-permission android:name="android.permission.ACCESS_GPS" />
  <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission
android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission
android:name="android.permission.READ_PHONE_STATE"></uses-permission>
  <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
permission>
```

Bibliotecas externas usadas

Para a implementação de algumas componentes da aplicação, recorreu-se à utilização de bibliotecas externas. Apresentam-se as bibliotecas utilizadas:

SlidingMenu library ¹⁰: A utilização desta biblioteca permite a criação do menu principal da aplicação, ou seja do *sliding* menu da aplicação. Este tipo de menu é cada vez mais usado nas aplicações Android e é usado em aplicações de companhias populares tais como *Youtube*, *Facebook* ou *Google*. A Figura 36 (figura mais à esquerda e figura do meio) mostra a utilização deste tipo de menus. Ao carregar no botão que se encontra no topo superior esquerdo ou deslizando com os dedos da esquerda para a direita, o ecrã desliza para abrir o menu principal .

FacebookSDK library ¹¹: A utilização desta biblioteca foi necessária para a criação da componente de partilha de informação na rede social *Facebook*. Trata-se de uma biblioteca oficial do *Facebook*, oferecendo as funcionalidades de *login*, *logout*, partilha de dados, efetuar *likes*, entre outras funcionalidades.

Twitter4J ¹²: A utilização desta biblioteca foi necessária para a criação da componente de partilha de informação na rede social *Twitter*. Após alguma investigação, verificou-se que não existe nenhuma biblioteca oficial *Twitter* para programadores, para o sistema operativo *Android*. No entanto, esta biblioteca pareceu ser a mais usada pela comunidade e considerada a mais estável e segura, utilizando OAuth para autenticação dos utilizadores.

TouchListView ¹³: Esta biblioteca foi utilizada para implementar a funcionalidade de *drag and drop* nos elementos de uma rota criada pelo utilizador, permitindo alterar a ordem dos pontos da rota, através de *drag and drop* dos elementos.

Gson: Esta biblioteca foi utilizada para efetuar o *parsing* das respostas Json vindas do servidor para a aplicação móvel ¹⁴.

Designs Patterns Android aplicadas

Uma característica importante para uma aplicação bem projetada, é que esta tenha uma navegação eficiente [67]. Segue-se uma lista de *designs patterns* utilizadas na aplicação, reconhecidas pela comunidade de desenvolvedores para Android.

Action Bar ¹⁵: Trata-se de um *design pattern* muito utilizada em aplicações Android. Consiste numa barra de ações que identifica a aplicação e fornece ao utilizador as possíveis ações e modos de navegação. A *Action Bar* deve ser usada nas atividades que necessitam apresentar ações aos utilizadores ou modos de navegação. Na Figura 36, é possível visualizar a utilização da Action Bar, que consiste na barra verde que se encontra no topo do ecrã, permitindo a abertura de menus de cada lado do ecrã.

Swipe Views ¹⁶: Geralmente, as aplicações são construídas de uma forma hierárquica. No entanto, existem casos de aplicações em que a navegação horizontal pode achatar hierarquias verticais e tornar o acesso a dados relacionados de maneira mais rápida e

¹⁰ <https://github.com/jfeinstein10/SlidingMenu>

¹¹ <https://developers.facebook.com/docs/getting-started/facebook-sdk-for-android/3.0/>

¹² <http://twitter4j.org/en/index.html>

¹³ <https://github.com/commonsguy/cwac-touchlist>

¹⁴ <http://code.google.com/p/google-gson/>

¹⁵ <http://developer.android.com/guide/topics/ui/actionbar.html>

¹⁶ <http://developer.android.com/design/patterns/swipe-views.html>

agradável. *Swipe views* permitem ao utilizador mover de forma eficiente de item para item usando um simples gesto tornando assim os dados de navegação e consumo numa experiência mais fluente [67].

Sincronização dos dados entre aplicação e servidor

A sincronização dos dados entre a aplicação e o servidor seguiu os esquemas apresentados na secção 4.2.7 da arquitetura. Para a transferência de imagens, que são conteúdos mais “pesados”, foi adotado o mecanismo que permite apenas transferir os conteúdos que foram alterados ou adicionados.

Caso o utilizador deseje proceder à atualização de conteúdos e o servidor se encontre indisponível por algum motivo, o utilizador será informado que não é possível proceder à atualização de conteúdos naquele momento e para tentar mais tarde. Caso o servidor falhe durante a atualização de dados, são repostos os dados existentes antes da atualização dos dados e o utilizador é informado para tentar novamente mais tarde a atualização dos dados. Aqui, era importante a aplicação não falhar caso ocorresse algum problema do lado de servidor.

Ecrãs da aplicação final

São apresentadas algumas capturas de ecrã da aplicação final. O Anexo III apresente todas as capturas de ecrã da aplicação assim como uma descrição de cada uma delas. A Figura 36 apresenta capturas de ecrã para a listagem das plantas (figura mais à esquerda), para o menu principal da aplicação (figura do meio) e para a utilização do mapa na aplicação (figura mais à direita). A listagem das plantas apresenta, por defeito, todas as plantas presentes na aplicação sob a forma de uma grelha. O menu principal é aberto ao carregar no botão presente no topo superior esquerdo ou então deslizando o dedo no ecrã da esquerda para a direita. O mapa apresenta todas as plantas presentes na aplicação através de pontos vermelhos. Ao carregar num dos pontos um pequeno painel informativo é aberto com um ícone da planta e o nome da mesma.

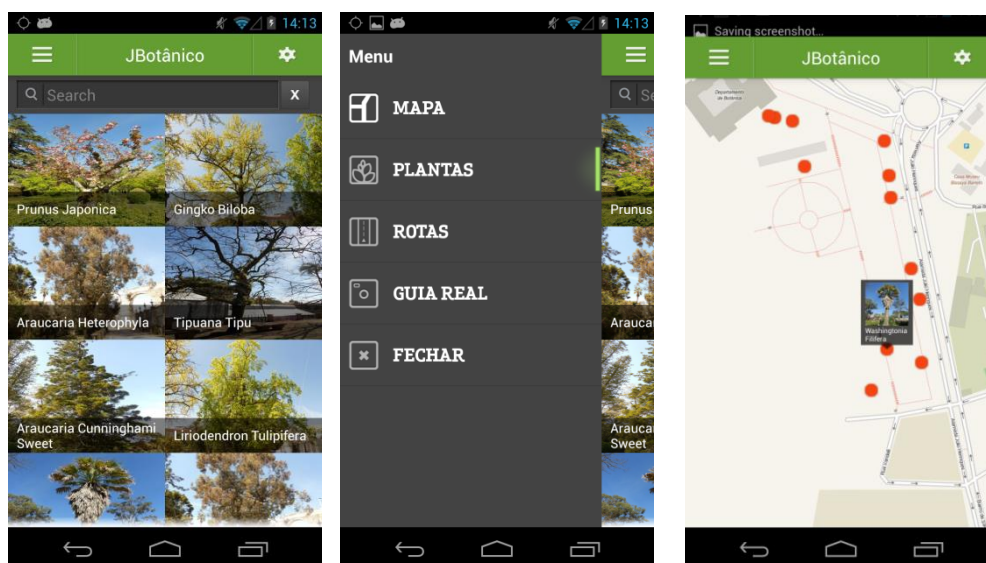


Figura 36: Resultado final para a listagem das plantas(figura da esquerda) e a utilização do mapa na aplicação (figura da direita)

A Figura 37 apresenta resultados para a utilização de realidade aumentada na aplicação móvel, permitindo a georreferenciação das plantas. Para cada planta, é mostrado um ícone da mesma, o nome assim que a distância até à mesma. A distância é alterada em tempo real à medida que o utilizador se desloca.



Figura 37: Representação de uma captura de ecrã para a utilização da realidade aumentada

6.2.3 Principais problemas encontrados e soluções

Apresentam-se nesta secção os principais problemas encontrados durante o desenvolvimento da aplicação móvel e, quando possível, a solução encontrada para contornar os mesmos.

Conteúdos multimédia para a aplicação móvel

A obtenção de conteúdos para introduzir na aplicação móvel foi logo o primeiro problema encontrado. Após uma reunião com representantes do Jardim Botânico, o autor ficou informado da existência de uma quantidade quase nula de dados sobre as plantas presentes no Jardim Botânico. Os representantes do Jardim mostraram dispostos a ajudar, no entanto não sabiam em que alturas teriam esses dados a serem disponibilizados. Torna-se aqui evidente que a falta de conteúdos oficiais pode limitar a qualidade da aplicação no que diz respeito aos conteúdos apresentados.

O autor decidiu portanto antecipar-se e não ficar dependente dos conteúdos que a entidade gestora do Jardim poderia ou não disponibilizar. Foi efetuado por parte do autor um levantamento de dados numa área específica do Jardim Botânico. O levantamento consistiu na recolha dos nomes de 20 plantas assim como fotografias. Finalmente, de modo a fornecer um máximo de informação, foram efetuadas pesquisas sobre cada uma das plantas para fornecer ao visitante uma descrição da mesma. A iniciativa de levantamento de dados tornou-se uma escolha acertada pois a entidade gestora do Jardim Botânico não foi capaz de fornecer quaisquer dados sobre as plantas presentes no recinto.

Realidade aumentada

Chagado o memento para o desenvolvimento da componente de realidade aumentada, verificou-se que a plataforma *Wikitude* oferecia uma versão grátis da plataforma mas somente de maneira experimental. A versão experimental apresenta os

inconvenientes de não deixar eliminar no ecrã da câmara o logo da *Wikitude* e, além disso, apresenta marcas de água no ecrã da câmara. É possível obter uma licença académica que permite eliminar as marcas de água assim como o logo da plataforma no entanto, não é possível publicar a aplicação no *market* de aplicações do *Android* com este tipo de licença. Licenças completas para esta plataforma não são gratuitas e requerem grandes custos adicionais.

Para resolver este problema, foi encontrada uma extensão da plataforma *Mixare*, identificada na secção 2.6.2 (realidade aumentada) do Estado da arte. A plataforma funciona sem necessidade de ligação á internet, requisito importante para esta componente.

Mapa

A implementação da componente de mapas, da aplicação, também levantou alguns problemas. Na elaboração dos requisitos, ficou definido que a aplicação deveria funcionar sem recorrer à rede internet. Ficou definida a utilização da biblioteca *OpenStreetMaps* uma vez que permite a utilização de mapas sem recorrer à internet. No entanto, aquando a implementação da componente de mapas, verificou-se que o nível máximo de Zoom no mapa é muito pequeno, devido a limitações impostas pela *OpenStreetMaps* em relação aos níveis de Zoom que é possível utilizar em mapas *offline*.

Uma possível solução para este problema, seria a utilização dos mapas, recorrendo à internet. No entanto, isto iria de encontro com o requisito de utilização do mapa sem recorrer à internet. Portanto, decidiu-se continuar com a utilização do mapa em modo *offline*, oferecendo um nível máximo de zoom mais limitado.

Google API

Tal como definido nos requisitos, a aplicação deveria ser capaz de interagir com a rede social *Google plus*, para permitir ao utilizador partilhar dados sobre a sua visita ao Jardim Botânico. No entanto, a integração com esta rede social não foi possível uma vez que o SDK foi modificado para ser apenas de leitura, não permitindo a introdução de dados por parte de utilizadores ¹⁷.

Códigos QR

Um dos requisitos definidos para a aplicação móvel era a possibilidade de leitura de códigos QR para aceder à informação detalhada de uma planta. Como já referido acima, o Jardim Botânico não foi capaz de fornecer material necessário para introduzia na aplicação. Consequentemente, também não foi possível obter o material necessário para a utilização de um leitor de códigos QR na aplicação móvel. Ficou portanto definido que o leitor de Códigos QR não seria implementado nesta versão da aplicação, podendo vir a ser implementado posteriormente.

¹⁷ <https://developers.google.com/+/api/>

6.4 Desenvolvimento da componente de visualização de estatísticas

Tal como definido inicialmente, a aplicação de visualização de estatísticas foi desenvolvida através da biblioteca *PolyMaps*, tratando-se de uma biblioteca javascript especializada para a representação de dados estatísticos. De modo a criar um *mini-site* para a representação dos dados estatísticos, a biblioteca Polymaps foi integrada num projeto da *Framework Django*, facilitando a construção do *mini-site*.

Através desta aplicação, é possível ao utilizador visualizar dados sobre algumas métricas recolhidas pela aplicação móvel para o Jardim Botânico de Coimbra. É possível visualizar o seguinte:

- Plantas mais/menos vistas pelos visitantes;
- Plantas que recolhem mais/menos *likes*;
- Plantas mais/menos partilhadas na rede social Facebook;
- Plantas mais/menos Partilhadas na rede social Twitter;
- Plantas que recolhem mais/menos comentários nas partilhas associadas às plantas.

Para além disso, para cada uma das métricas recolhidas, o utilizador pode escolher aplicar filtros temporais à visualização dos resultados. Distinguem-se aqui dois tipos de filtros:

- Filtro temporal num intervalo de tempo do dia: Por exemplo, entende-se por aqui poder visualizar as visitas que ocorrem das 13 horas até as 15 horas durante o dia. Recorrendo a este tipo de filtro, é possível saber quais as alturas do dia onde ocorrem mais visitas.
- Filtro temporal num intervalo de tempo em dias/mês: Por exemplo, entende-se aqui poder visualizar as visitas que ocorreram do dia 15 de Agosto até ao dia 20 de Setembro. Este tipo de filtro permite verificar o numero de visitas totais num determinado intervalo de tempo. Este filtro pode ser bastante útil uma vez que, escolhendo as datas para tratamento dos resultados, pode-se criar resultados para muitas outras métricas tais que comparar os resultados obtidos para cada mês, obter resultados em função das estações do ano, entre outras métricas possíveis de criar.
- Filtro temporal que combina os dois anteriores. Por exemplo, um utilizador pode escolher visualizar as visitas que ocorreram do dia 15 de Agosto até ao dia 22 de Novembro, entre as 15 horas e 32 minutos e as 17 horas e 30 minutos. Com isto, é possível comparar resultados num espaço de tempo durante o dia, em intervalos de tempo de dias diferentes.

6.4.1 Estrutura do código da aplicação

Foi apresentada na secção 6.1 a estrutura que segue tipicamente uma aplicação desenvolvida em Django. Assim, dado que a aplicação para visualização de estatísticas está desenvolvida em Django, é aqui necessário voltar a descrever a estrutura do código da

mesma. Apenas são aqui apresentados elementos novos em relação ao apresentado na secção 6.1

Assim, a aplicação segue a seguinte estrutura:

```
mysite/  
  manage.py  
  mysite/  
    __init__.py  
    settings.py  
    urls.py  
    wsgi.py
```

, tratando-se da estrutura base da aplicação ou seja, ficheiros de configuração. Dentro desta pasta é criada a aplicação em si para poder visualizar as estatísticas, tendo a seguinte estrutura:

```
polls/  
  static  
  templates  
  __init__.py  
  admin.py  
  models.py  
  tests.py  
  views.py
```

Aqui, destacam-se duas novas pastas, em relação à estrutura apresentada para a criação da API:

- **Static:** Esta pasta é responsável por manter todos os ficheiros para formatação das páginas *web*, tais como ficheiros de javascript, css ou imagens. É nesta pasta que são também adicionados os ficheiros de configuração da biblioteca Polymaps.
- **Templates:** Esta pasta é responsável por manter os ficheiros html que são utilizados na aplicação, para apresentar os elementos no browser.

6.4.2 Capturas de ecrã da aplicação

É apresentada uma captura de ecrã da aplicação desenvolvida. O Anexo V apresenta as capturas de ecrãs da aplicação de modo a visualizar o resultado final da mesma assim como comentários para explicar os resultados presentes nas imagens capturadas.

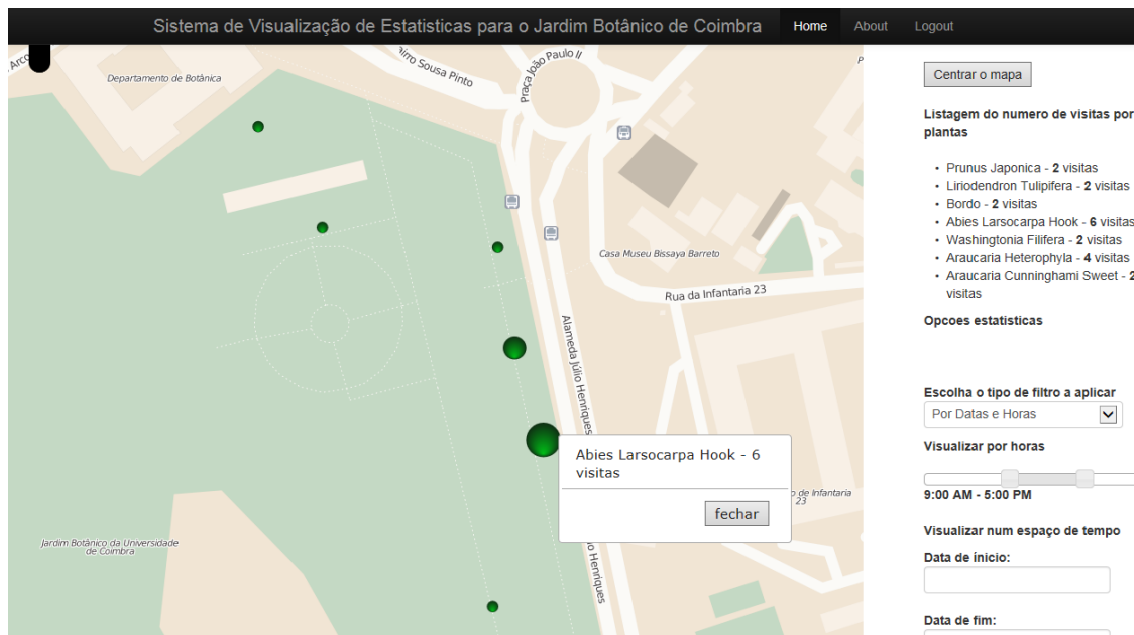


Figura 38: Resultado para a representação de dados estatísticos na aplicação final.

A Figura 38 mostra a representação dos dados para uma determinada métrica. Todas as métricas seguem o mesmo modelo de representação dos dados. Através de um mapa e de esferas de cor verde para representar as plantas na sua localização, são apresentados os dados recolhidos pela aplicação. Do lado direito da página encontra-se o menu que efetua a listagem das plantas indicando o nome e número de visitas para cada uma. Por baixo da listagem, existe um menu com os filtros que podem ser aplicados aos dados.

6.5 Outras aplicações desenvolvidas no decorrer do estágio

Além da aplicação móvel para o Jardim Botânico, o autor participou no desenvolvimento de uma outra aplicação móvel. A aplicação consiste num guia móvel para um ponto turístico de Aveiro. Por questões de privacidade não é possível divulgar neste momento o ponto turístico para o qual foi criada a aplicação. A aplicação permite ao utilizador entrar na história de uma das personagens pertencente a esse ponto de interesse e seguir um determinado percurso, visualizando informação e conteúdos multimédia sobre pontos de interesse presentes no percurso. Também é possível ao utilizador tirar as suas próprias fotografias durante a visita, e visualizá-las no diário de bordo da aplicação. Um ponto importante aqui é a qualidade e quantidade de informação usada na aplicação, fornecida diretamente por um designer ligado ao projeto. Assim, foi possível inserir muitos conteúdos na aplicação, provenientes do designer. Apresenta-se uma listagem das funcionalidade implementadas na aplicação:

- **Splash Screen:** Ao abrir a aplicação, é mostrada uma pequena animação introdutória.
- **Vídeo introdutório:** Após a pequena animação inicial, a aplicação inicia um vídeo introdutório para dar ao utilizador informações sobre o ponto turístico em questão. O utilizador pode escolher terminar o vídeo a qualquer momento.

- **Listagem dos percursos:** Uma vez visualizado o vídeo introdutório, o utilizador pode escolher um percurso a seguir, numa listagem de percursos criados cada um deles à partir da vida de uma das personagens da antiga aldeia. A Figura 67 presente no Anexo IV mostra capturas de ecrã desta componente.
- **Visualização do BI da personagem escolhida:** Uma vez escolhido o percurso, o utilizador acede ao BI da personagem escolhida na lista de percursos. A Figura 68 presente no Anexo IV mostra capturas de ecrã desta componente.
- **Seguir um percurso no mapa:** após visualizar o BI da personagem escolhida na lista de percursos, o utilizador pode escolher iniciar o percurso, passando para um mapa com os diferentes pontos de interesse pertencentes ao percurso. A Figura 69 presente no Anexo IV mostra capturas de ecrã desta componente.
- **Visualizar conteúdos multimédia de um ponto de interesse:** O utilizador pode visualizar conteúdos multimédia sobre um determinado ponto de interesse. Além de poder visualizar uma galeria com várias imagens, o utilizador pode aceder a conteúdos de áudio. Também é possível ao utilizador tirar as suas próprias fotografias sobre o ponto de interesse, podendo serem visualizadas posteriormente no diário de bordo. A figura 70 presente no Anexo IV mostra capturas de ecrã desta componente.
- **Visualizar diário de bordo:** O utilizador pode visualizar as fotografias que tirou a cada um dos pontos de interesse. As fotografias são agrupadas por ponto de interesse. A figura 71 presente no Anexo IV mostra capturas de ecrã desta componente.

O Anexo IV apresenta capturas de ecrã da aplicação desenvolvida assim como uma breve explicação sobre os mesmos.

Escrita de um artigo para revista científica

O autor participou também na escrita de um artigo para submissão na revista RISTI¹⁸ (Revista Ibérica de Sistemas e Tecnologias de Informação). A RISTI é um periódico científico, propriedade da AISTI (Associação Ibérica de Sistemas de e Tecnologias de Informação), que foca a investigação e a aplicação prática inovadora no domínio dos sistemas e tecnologias de informação. Cada número da RISTI aborda uma temática diferente. Para este número, onde foi submetido um artigo, a temática era dedicada à Interação Humano-Computador relacionado com tópicos tais que Usabilidade, Acessibilidade, Desenvolvimento de Interfaces, Realidade Virtual, Aumentada, entre outros tópicos.

O artigo foi escrito apresentando a aplicação desenvolvida para o Jardim Botânico de Coimbra, apresentando a aplicação e suas funcionalidades mas focando principalmente a utilização da realidade aumentada na aplicação. No entanto, o artigo não foi aprovado para publicação. Cada número da RISTI costuma aceitar somente 5 a 8 artigos para publicação. A taxa média de aceitação é bastante apertada, situando-se abaixo dos 15%. No entanto, apesar de o artigo não ter sido aceite, a revisão do mesmo deixou bons comentários por parte dos revisores quanto ao produto apresentado.

¹⁸ http://www.aisti.eu/index.php?option=com_content&task=view&id=20

Capítulo 7

Testes

Este capítulo apresenta o plano de testes seguido tendo em vista a validação dos resultados obtidos para a aplicação móvel. Os testes são apresentados, divididos por testes para a componente servidor, testes para a aplicação móvel e testes para a componente de visualização de estatísticas.

7.1 Testes para o servidor

A ferramenta *Tastypie* fornece algumas ferramentas que trabalham sobre a camada de suporte de testes do Django, na forma de um *TestApiClient* e *ResourceTestCase* especializados.

O *ResourceTestCase* trabalha sobre o Django *TestCase*. Ele fornece uma quantidade de métodos de verificação extra que são específicos para APIs e utiliza o *TestApiClient* para realizar os pedidos corretamente. Trata-se aqui de testes automatizados.

Uma vez que alguns métodos são utilizados da API utilizada em grande parte dos projetos da empresa, apenas foram testados os métodos que foram criados de raiz. São apresentados a seguir os testes criados para testar os novos métodos adicionados à API.

Modulo 1: Coordinates

Cenário: Test-01 – Teste à inserção de conteúdos

Pré-condições: Base de dados vazia

Fluxo do teste:

1. Verificar a quantidade de entradas na tabela onde são guardadas as coordenadas;
2. Fazer uma chamada ao método POST para inserir um objeto coordenada com os seguintes parâmetros: latitude, longitude e data e longitude;
3. Verificar que a quantidade de entradas da tabela que guarda as coordenadas foi aumentada de uma unidade relativamente ao resultado obtido no ponto 1.
4. Verificar que o elemento inserido foi de facto inserido.

Cenário: Test-02 – Teste à eliminação de conteúdos

Pré-condições:

1. A tabela que guarda as coordenadas conter pelo menos um elemento;
2. A tabela que guarda as coordenadas conter o elemento que se deseja eliminar.

Fluxo do teste:

1. Verificar a quantidades de entradas na base de dados, efetuando uma contagem;
2. Eliminar uma entrada na base de dados chamando o método *DELETE* com os seguintes parâmetros: *id* (identificador) do elemento que se deseja eliminar;
3. Verificar que a quantidade de entradas na base de dados diminui de uma unidade relativamente ao resultado obtido no ponto 1, efetuando uma contagem dos elementos presentes na tabela.
4. Verificar que o elemento eliminado já não se encontra na base de dados;

Cenário: Test-03 – Teste à listagem/obtenção de conteúdos

Fluxo do teste:

1. Efetuar uma listagem dos conteúdos presentes na tabela de coordenadas, chamando o método GET;
2. Verificar que o pedido devolveu uma resposta com *response code* 201 assim como uma lista com os elementos presentes na tabela da base de dados.

Módulo 2: Social_Network

Uma vez que este módulo é implementado para guardar as atividades relacionadas com as redes sociais, pretende-se efetuar testes com fim a verificar se as partilhas nas redes sociais são bem guardadas na base de dados (POST), eliminados (DELETE) ou então listados/obtidos para o tratamento estatístico. (GET).

Facebook_shares/

Cenário: Test-05 – Teste à obtenção/listagem dos conteúdos

Pré-condições:

1. A tabela que guarda as partilhas de conteúdos na rede social *Facebook* conter pelo menos um elemento.

Fluxo do teste:

1. Efetuar uma listagem dos objetos presentes nesta tabela que guarda as partilhas de conteúdos na rede social *Facebook*, efetuando uma chamada ao método GET;
2. Verificar que o pedido devolveu uma resposta com *response code* 201 assim como uma lista com os elementos presentes na tabela.

Cenário: Test-06 – Teste à inserção de conteúdos

Fluxo do teste:

1. Verificar a quantidade de objetos na base de dados, efetuando uma contagem;
2. Adicionar uma entrada na base de dados, efetuando uma chamada ao método POST;
3. Verificar que a quantidade de objetos foi incrementada de uma unidade relativamente ao valor obtido no ponto 1.
4. Verificar que o elemento foi mesmo inserido, efectuado uma chamada ao método GET com o seguinte parâmetro: id(identificador) do elemento inserido.

Cenário: Test-07 – Teste à eliminação dos conteúdos

Pré-condições:

1. A tabela que guarda as operações de partilhas no *Facebook* conter pelo menos um elemento;
2. A tabela que guarda as operações da partilhas no *Facebook* conter o elemento que se deseja eliminar.

Fluxo do teste:

1. Verificar a quantidade de objetos na base de dados, efetuando uma contagem;
2. Eliminar uma entradas na base de dados, recorrendo à chamada do método DELETE com o seguinte parâmetro: id (identificador) do elemento a eliminar;
3. Verificar que a quantidade de objetos foi decrementada de acordo com a quantidade de objetos eliminados.
4. Verificar que o elemento já não se encontra na base de dados, efectuado uma chamada ao método GET com o parâmetro do id (identificador).

Twitter_shares/

Cenário: Test-08 – Teste à obtenção/listagem dos conteúdos

Cenário semelhante ao cenário Test-05 – Teste à obtenção/listagem dos conteúdos. No entanto o teste é aplicado às partilhas da rede social Twitter.

Cenário: Test-09 – Teste à inserção de conteúdos

Smelhante ao cenário Test-06 – Teste à inserção de conteúdos. No entanto aplicado às partilhas na rede social Twitter.

Cenário: Test-10 – Teste à eliminação de conteúdos

Cenário semelhante ao cenário Test-07 – Teste à eliminação dos conteúdos. No entanto, o teste é aplicado às partilhas na rede social Twitter.

Módulo 3: Resources_History

Uma vez que este módulo é implementado para guardar o histórico das visitas relacionadas com qualquer planta, pretende-se efetuar testes com fim a verificar se os

recursos são bem guardados na base de dados (POST), eliminados (DELETE) ou então listados/obtidos para o tratamento estatístico (GET).

1. Resources_History/

Cenário: Test-11 – Teste à obtenção/listagem dos conteúdos

Cenário semelhante ao cenário Test-05 – Teste à obtenção/listagem dos conteúdos. No entanto, o cenário é aplicado ao histórico de visualização das plantas.

Cenário: Test-12 – Teste à inserção de conteúdos

Cenário Semelhante ao cenário Test-06 – Teste à inserção de conteúdos. No entanto este cenário é aplicado ao histórico de visitas das plantas.

Cenário: Test-13 – Teste à eliminação de conteúdos

Cenário semelhante ao cenário Test-07 – Teste à eliminação dos conteúdos. No entanto, este teste é aplicado ao histórico de visitas das plantas.

Módulo 7: Likes

Uma vez que este módulo é implementado para guardar os *likes* efetuados a conteúdos multimédia, plantas ou rotas, para posteriormente servirem para tratamento estatístico, pretende-se efetuar testes com fim a verificar se os recursos são bem guardados na base de dados (POST), eliminados (DELETE) ou listados/obtidos para inserção na aplicação (GET).

1. Likes_Resources/

Cenário: Test-14 – Teste à listagem de todos os conteúdos

Cenário semelhante ao cenário Test-05 – Teste à obtenção/listagem dos conteúdos. No entanto, este teste é aplicado aos likes das plantas.

Cenário: Test-15 – Teste à inserção de conteúdos

Cenário semelhante ao cenário Test-06 – Teste à inserção de conteúdos. No entanto, este teste é aplicado aos likes das plantas.

Cenário: Test-16 – Teste à eliminação de conteúdos

Cenário semelhante ao cenário Test-07 – Teste à eliminação dos conteúdos. No entanto, este teste é aplicado aos *likes* das plantas.

Segue-se uma tabela com os testes que foram definidos e os resultados obtidos. Foram feitas várias iterações de testes, até chegar a uma fase onde todos os testes

passavam com sucesso. São apresentados na Tabela 22 os resultados para a iteração de testes executados. .

Iteração 1

Teste	Resultado	Observações
Teste 1	Passou	
Teste 2	Passou	
Teste 3	Passou	
Teste 4	Passou	
Teste 5	Passou	
Teste 6	Passou	
Teste 7	Passou	
Teste 8	Passou	
Teste 9	Passou	
Teste 10	Passou	
Teste 11	Passou	
Teste 12	Passou	
Teste 13	Passou	
Teste 14	Passou	
Teste 15	Passou	
Teste 16	Passou	

Tabela 22: Resultados para a iteração de testes para o servidor

7.2 Testes para a aplicação móvel

De modo a testar a aplicação móvel, foram definidos testes de usabilidade, testes em ambiente real e testes aos mecanismos de sincronização de dados. Segue-se a descrição e resultados obtidos para cada tipo de testes.

7.2.1 Testes de usabilidade

Testes de usabilidade são necessários para identificar falhas e melhorias numa aplicação deste tipo. Uma das soluções encontradas para testar o grau de satisfação de utilizadores assim como a usabilidade da aplicação, consistiu na realização de testes recorrendo *System Usability Scale* (SUS), que consiste num questionário de 10 perguntas, cada uma com uma escala de respostas de 5 categorias [68] O resultado final obtido aplicando esse questionário permite determinar o grau de qualidade da aplicação em termos de usabilidade. Segue-se uma listagem das perguntas presentes no questionário:

1. Penso que gostaria de usar este sistema com frequência;
2. Encontro o sistema desnecessariamente complexo;
3. Acho que o sistema era fácil de usar;
4. Acho que seria necessária o apoio de uma pessoa para ser capaz de usar o sistema;
5. Acho que as diversas funções deste sistema foram bem integradas;
6. Achei que existiam muitas inconsistências no sistema;
7. Consigo imaginar a maioria das pessoas aprenderem a usar este sistema muito rapidamente;
8. Acho o sistema muito complicado de usar;
9. Senti-me muito confiante ao utilizar o sistema;
10. Precisava aprender um monte de coisas antes de poder utilizar este sistema.

A Figura 39 representa o formato de resposta às perguntas do questionário SUS:

Strongly Disagree 1	2	3	4	Strongly Agree 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 39: formato de resposta do SUS

Pontuação dos resultados:

- Para os itens ímpares: subtrair um valor à resposta do utilizador;
- Para os itens pares: subtrair a 5 valores ao valor da resposta do utilizador;
- Isto permite escalar todos os resultados de 0 a 4(sendo 4 a resposta mais positiva);
- Adicionar as respostas convertidas para cada utilizador e multiplicar esse total por 2.5. Isto converte a gama de possíveis valores de 0 a 100 em vez de estar de 0 a 40.

Outras características do SUS [68]:

- As pontuações dos SUS não são percentagens: Apesar da pontuação SUS variar entre 0 e 100, o resultado não corresponde a uma percentagem. Embora seja tecnicamente correto que uma pontuação SUS de 70 em 100 represente cerca de 70% da pontuação máxima possível, é sugerido que a pontuação está no percentil 100%. A pontuação nesse nível significa que a aplicação testada está acima da média em termos de usabilidade. SUS permitem medir a usabilidade e a aprendizagem: Enquanto SUS foi apenas destinado a medir a facilidade de uso (apenas uma dimensão), uma pesquisa recente mostra que ele também fornece uma medida global de satisfação do sistema assim como subescalas de usabilidade e capacidade de aprendizagem. Os itens 4 e 10 fornecem a dimensão responsável pela capacidade de aprendizagem e os outros 8 itens fornecem a dimensão da usabilidade.
- SUS é de confiança: Confiabilidade refere-se à forma como os utilizadores respondem aos itens (repetibilidade das respostas). SUS tem-se mostrado ser mais confiável e deteta diferenças em tamanhos de amostras menores do que questionários home-grown e outros disponíveis no mercado. O tamanho da amostra e a confiabilidade não estão ligados um ao outro portanto, o SUS pode ser usado em tamanhos de amostras muito pequenos (tão poucos como dois utilizadores) e ainda assim gerar resultados fiáveis.
- SUS é válido: Validade refere-se à forma como alguma coisa pode ser medida e aquilo que é pretendido medir. Neste caso, é a usabilidade que é pretendida medir.
- SUS não é diagnóstico: SUS não se destina a diagnosticar problemas de usabilidade . Apenas dá uma classificação do sistema sobre a usabilidade.

Resultados obtidos e interpretação:

Após recolha e análise das respostas obtidas, foi obtida uma classificação de 80. 12 utilizadores responderam ao questionário. Para um teste de usabilidade deste tipo (SUS), a classificação média definida é de 68. Qualquer resultado abaixo desse valor é considerado abaixo da média e qualquer resultado acima desse valor é considerado acima da média. Portanto, o resultado obtido (80) é considerado acima da média. Com base no tipo de teste executado e no resultado obtido, pode-se considerar que a aplicação oferece bons resultados quanto à usabilidade da aplicação desenvolvida.

7.2.2 Testes em ambiente real

Testes em ambiente real são necessários para visualizar e perceber o comportamento da aplicação ao ser usada no interior do Jardim Botânico de Coimbra. Através deste tipo de teste, é possível descobrir possíveis falhas que a aplicação apresenta e posteriormente proceder a resolução dos problemas encontrados.

Segue-se uma tabela com os testes que foram definidos e os resultados obtidos. Foram feitas várias iterações de testes, até chegar a uma fase onde todos os testes passavam com sucesso. São apresentados os resultados para cada uma das iterações.

Iteração 1:

Equipamento utilizado: Samsung Galaxy Nexus

Teste	Descrição	Resultado
Teste_1: Aceder à aplicação	Testa se a aplicação é capaz de arrancar sem dar problemas	Passou
Teste 2: Atualização de conteúdos	Testa se a aplicação é capaz de detetar uma ligação à internet e proceder à atualização dos dados corretamente	Passou
Teste_3: Mapa do Jardim	Testa se é possível aceder ao mapa do Jardim e visualizar no mesmo as plantas presentes no Jardim	Falhou
Teste_4: Posicionamento do visitante	Testa se o utilizador consegue visualizar a sua posição no mapa da aplicação, à medida que se vai deslocando.	Falhou
Teste_5: Aceder à descrição de uma planta a partir do mapa	O utilizador carrega numa das plantas presente no mapa e acede à descrição da mesma.	Falhou
Teste_6: Listagem das plantas	Teste se o utilizador consegue aceder a uma listagem completa das plantas presentes no jardim.	Passou
Teste_7: Listagem das plantas em função da posição (filtro de proximidade)	Testa se o filtro de listagem das plantas em função da distancia a que se encontra. Funciona corretamente.	Passou
Teste_8: Listagem das plantas por ordem alfabética	Testa se o filtro de listagem das plantas por ordem alfabética funciona corretamente	Passou

Teste_9: Listagem das plantas em grelha	O utilizador escolhe a opção de visualizar a listagem das plantas sob a forma de grelha.	Passou
Teste_10: Listagem das plantas em lista	O utilizador escolhe a opção de visualizar a listagem das plantas sob a forma de lista.	Passou
Teste_11: Aceder à descrição de uma planta a partir da listagem	Testa se o utilizador consegue escolher uma das plantas presentes na listagem e visualizar os detalhes da mesma.	Passou
Teste_12: Visualizar plantas presentes no Jardim através da realidade aumentada	Testa se o utilizador consegue aceder ao guia real da aplicação, visualizando as plantas existentes no interior do Jardim assim como a distância até à mesma.	Passou
Teste_13: Atualização das distâncias até às plantas nos modos de realidade aumentada.	Testa se , nos modos de realidade aumentada, a distância apresentada até as plantas é atualizada à medida que o utilizador se desloca.	Passou
Teste_14: Visualizar detalhes de uma planta a partir da realidade aumentada	Testa se o utilizador, ao carregar numa das plantas presentes no guia real, consegue aceder à informação detalhada da mesma.	Passou
Teste_15: Pesquisa de uma planta	O utilizador acede à barra de pesquisa presente na listagem das plantas e pesquisa por uma determinada planta. A listagem deve apresentar os resultados correspondentes à pesquisa. À medida que é alterada a pesquisa, os resultados são alterados em tempo real.	Passou
Teste_16: Partilha de dados na rede social Facebook	Testa se ao o utilizador escolher a opção de partilhar dados na rede social do Facebook, estes são partilhados com sucesso	Passou
Teste_17: Partilha de dados na rede social Twitter	Testa se ao o utilizador escolher a opção de partilhar dados na rede social Twitter, estes são partilhados com sucesso.	Passou
Teste_18: Listagem das rotas	Testa se as rotas são listadas corretamente, diferenciando as rotas pré-definidas e as rotas criadas pelo utilizador.	Passou
Teste_19: Seguir uma rota no mapa 2D	O utilizador escolhe uma rota e decide segui-la através do mapa 2D. A aplicação deve ser capaz de indicar qual é o próximo ponto, assim como a distância até ao mesmo.	Falhou
Teste_20: Atualização dos pontos da rota no mapa 2D	Testa se à medida que é visitado um ponto da rota no mapa 2D, o próximo ponto é marcado como tal no mapa, os pontos visitados são devidamente marcados e os pontos que faltam visitar também devidamente marcados (marcadores diferentes)	Falhou
Teste_21: Seguir uma rota através da realidade	O utilizador escolhe uma rota e decide segui-la através da realidade aumentada. A aplicação deve ser capaz de indicar qual é o próximo	Passou

aumentada	ponto assim como a distância até ao mesmo.	
Teste_22: Atualização dos pontos da rota na realidade aumentada	Testa se à medida que é visitado um ponto da rota com a realidade aumentada, o próximo ponto é marcado como tal no mapa, os pontos visitados são devidamente marcados e os pontos que faltam visitar também devidamente marcados (marcadores diferentes)	Passou
Teste_23: Encontrar uma planta através do mapa 2D.	O utilizador está na visualização de detalhes de uma planta e decide encontra-la no mapa 2D.	Falhou
Test_24: Criação de uma rota pelo utilizador	Testa a criação de uma nova rota por parte do utilizador. O utilizador escolhe os pontos que deseja para constituir a rota e escolhe a ordem de visita dos pontos.	Passou
Test_25: Encontrar uma planta através do guia real	O utilizador está na visualização de detalhes de uma planta e decide encontra-la através do guia real.	Passou
Teste_26: Criar a sua própria rota	O utilizador escolhe a opção de criar a sua própria rota.	Passou

Tabela 23: Resultados para a primeira iteração de testes em ambiente real para a aplicação móvel

Na primeira iteração dos testes em ambiente real, alguns dos testes não foram capazes de passar com sucesso. Todos os teste que falharam estão ligados à utilização do mapa 2D da aplicação. Sempre que era necessário utilizar o mapa 2D, a aplicação falhava.

Após análise do código, foram feitas alterações para tentar resolver os “bugs” que impediram alguns testes de passar na iteração 1 de testes. Seguem-se os resultados obtidos para a iteração 2 de testes em ambiente real.

Iteração 2:

Equipamento utilizado: Samsung Galaxy Nexus

Teste	Descrição	Resultado
Teste_1: Aceder à aplicação	Testa se a aplicação é capaz de arrancar sem dar problemas	Passou
Teste 2: Atualização de conteúdos	Testa se a aplicação é capaz de detetar uma ligação à internet e proceder à atualização dos dados corretamente	Passou
Teste_3: Mapa do Jardim	Testa se é possível aceder ao mapa do Jardim e visualizar no mesmo as plantas presentes no Jardim	Passou
Teste_4: Posicionamento do visitante	Testa se o utilizador consegue visualiza a sua posição no mapa da aplicação, à medida que se vai deslocando.	Passou
Teste_5: Aceder à descrição de uma planta a partir do mapa	O utilizador carrega numa das plantas presente no mapa e acede à descrição da mesma.	Passou
Teste_6: Listagem das plantas	Teste se o utilizador consegue aceder a uma listagem completa das plantas presentes no jardim.	Passou
Teste_7: Listagem	Testa se o filtro de listagem das plantas em	Passou

das plantas em função da posição (filtro de proximidade)	função da distancia a que se encontra. Funciona corretamente.	
Teste_8: Listagem das plantas por ordem alfabética	Testa se o filtro de listagem das plantas por ordem alfabética funciona corretamente	Passou
Teste_9: Listagem das plantas em grelha	O utilizador escolhe a opção de visualizar a listagem das plantas sob a forma de grelha.	Passou
Teste_10: Listagem das plantas em lista	O utilizador escolhe a opção de visualizar a listagem das plantas sob a forma de lista.	Passou
Teste_11: Aceder à descrição de uma planta a partir da listagem	Testa se o utilizador consegue escolher uma das plantas presentes na listagem e visualizar os detalhes da mesma.	Passou
Teste_12: Visualizar plantas presentes no Jardim através da realidade aumentada	Testa se o utilizador consegue aceder ao guia real da aplicação, visualizando as plantas existentes no interior do Jardim assim como a distância até à mesma.	Passou
Teste_13: Atualização das distâncias até às plantas nos modos de realidade aumentada.	Testa se , nos modos de realidade aumentada, a distância apresentada até as plantas é atualizada à medida que o utilizador se desloca.	Passou
Teste_14: Visualizar detalhes de uma planta a partir da realidade aumentada	Testa se o utilizador, ao carregar numa das plantas presentes no guia real, consegue aceder à informação detalhada da mesma.	Passou
Teste_15: Pesquisa de uma planta	O utilizador acede à barra de pesquisa presente na listagem das plantas e pesquisa por uma determinada planta. A listagem deve apresentar os resultados correspondentes à pesquisa. À medida que é alterada a pesquisa, os resultados são alterados em tempo real.	Passou
Teste_16: Partilha de dados na rede social Facebook	Testa se ao o utilizador escolher a opção de partilhar dados na rede social do Facebook, estes são partilhados com sucesso	Passou
Teste_17: Partilha de dados na rede social Twitter	Testa se ao o utilizador escolher a opção de partilhar dados na rede social Twitter, estes são partilhados com sucesso.	Passou
Teste_18: Listagem das rotas	Testa se as rotas são listadas corretamente, diferenciando as rotas pré-definidas e as rotas criadas pelo utilizador.	Passou
Teste_19: Seguir uma rota no mapa 2D	O utilizador escolhe uma rota e decide segui-la através do mapa 2D. A aplicação deve ser capaz de indicar qual é o próximo ponto, assim como a distância até ao mesmo.	Passou
Teste_20: Atualização dos	Testa se à medida que é visitado um ponto da rota no mapa 2D, o próximo ponto é marcado	Passou

pontos da rota no mapa 2D	como tal no mapa, os pontos visitados são devidamente marcados e os pontos que faltam visitar também devidamente marcados (marcadores diferentes)	
Teste_21: Seguir uma rota através da realidade aumentada	O utilizador escolhe uma rota e decide segui-la através da realidade aumentada. A aplicação deve ser capaz de indicar qual é o próximo ponto assim como a distância até ao mesmo.	Passou
Teste_22: Atualização dos pontos da rota na realidade aumentada	Testa se à medida que é visitado um ponto da rota com a realidade aumentada, o próximo ponto é marcado como tal no mapa, os pontos visitados são devidamente marcados e os pontos que faltam visitar também devidamente marcados (marcadores diferentes)	Passou
Teste_23: Encontrar uma planta através do mapa 2D.	O utilizador está na visualização de detalhes de uma planta e decide encontra-la no mapa 2D.	Passou
Test_24: Criação de uma rota pelo utilizador	Testa a criação de uma nova rota por parte do utilizador. O utilizador escolhe os pontos que deseja para constituir a rota e escolhe a ordem de visita dos pontos.	Passou
Test_25: Encontrar uma planta através do guia real	O utilizador está na visualização de detalhes de uma planta e decide encontra-la através do guia real.	Passou
Teste_26: Criar a sua própria rota	O utilizador escolhe a opção de criar a sua própria rota.	Passou

Tabela 24: Resultados para a segunda iteração de testes em ambiente real para a aplicação móvel

7.2.3 Testes à precisão da realidade aumentada

Durante a realização dos testes em ambiente real, foi prestada particular atenção aos resultados obtidos para a componente de realidade aumentada. Era importante aqui que a aplicação fosse capaz de indicar o posicionalmente de uma planta com a maior precisão possível. Na melhor das hipóteses, o ícone representativo de uma planta deveria coincidir por completo com a planta correspondente.

Como é óbvio, os resultados obtidos dependem da precisão dos dados recolhidos relativos às coordenadas GPS de uma planta, bem como da capacidade de precisão do GPS do dispositivo móvel. Verificou-se que, em média, podia existir um erro de precisão de 3 ou 4 metros relativamente aos dados apresentados pela aplicação.

7.2.4 Testes aos mecanismos de sincronização de dados

Foram efetuados testes para verificar o funcionamento dos mecanismos de sincronização de conteúdos quando a aplicação móvel recebe dados do servidor, mas também nos casos em que a aplicação envia dados recolhidos pela mesma, para tratamento estatístico. Assim, foram definidos alguns cenários de teste, apresentados abaixo.

Cenário 1: Teste à transferência dos dados do servidor para a aplicação móvel

- Inicialmente, a base de dados da aplicação móvel está vazia;

- O utilizador escolhe a opção de proceder à sincronização dos dados;
- A sincronização dos dados, do servidor para a aplicação móvel é iniciada;
- A aplicação recebe os dados, e trata de guarda-los na base de dados local, sem a ocorrência de qualquer erro.
- A sincronização termina sem ocorrer qualquer erro que provoque a falha da aplicação.

Cenário 2: Teste à transferência dos dados entre o servidor e a aplicação com falha do servidor

- Inicialmente, a base de dados da aplicação móvel já contém alguns dados;
- O utilizador escolhe a opção de proceder a sincronização dos dados;
- A sincronização é iniciada e começam a ser transferidos dados do servidor para a aplicação móvel;
- O servidor é mandado a baixo durante a transferência dos dados.
- O utilizador é informado que não é possível proceder à sincronização dos dados, para tentar mais tarde.
- São repostos os dados presentes na aplicação antes do inicio da sincronização de dados.

Cenário 3: Teste a sincronização das imagens

- Inicialmente, a base de dados da aplicação móvel contém dados provenientes de uma sincronização anterior;
- O utilizador escolhe a opção de proceder à sincronização dos dados;
- Aquando o momento da sincronização das imagens, o servidor envia para a aplicação a lista das imagens presentes no servidor;
- A aplicação móvel verifica quais as imagens que já contém, aquelas que já não existem do lado do servidor e aquelas que foram adicionadas;
- A aplicação móvel faz pedidos ao servidor pelas imagens que não existem na base de dados da aplicação;
- A aplicação elimina as imagens que já não existem;

Cenário 4: Teste ao envio de dados da aplicação móvel para o servidor

- Inicialmente, a aplicação móvel contém dados recolhidos para estatísticas;
- A aplicação móvel deteta que existe ligação à rede móvel e procede ao envio dos dados para o servidor;
- A aplicação móvel apaga da sua base de dados os dados que foram enviados para o servidor;

A tabela 25 apresenta os resultados obtidos para a iteração de testes. Todos os testes passaram com sucesso.

Iteração 1

Cenário	Resultado	Observações
Cenário 1	Passou	
Cenário 2	Passou	
Cenário 3	Passou	
Cenário 4	Passou	

Tabela 25: Resultados para a iteração de testes para a sincronização dos dados

7.3 Testes para a aplicação de visualização de estatísticas

Uma série de testes foram definidos de modo a verificar o funcionamento correto da aplicação de visualização de estatísticas. Para isso, foram criados vários cenários que são apresentados a seguir. Os cenários apresentados foram testados efectuando variações nos valores dos filtros.

Cenário 1: Teste ao número de visitas das plantas

- Inicialmente, a base de dados de visitas está vazia;
- São efetuadas duas visitas à planta X e cinco visitas à planta Y através da aplicação móvel;
- Os dados são enviados da aplicação móvel para o servidor.
- O utilizador acede à aplicação de visualização de estatísticas e escolhe a opção de visualizar o número de visitas às plantas.
- O utilizador verifica que são apresentadas duas visitas para a planta X e cinco visitas para a planta Y.

Cenário 2: Teste ao número de *Likes* associados às plantas

- Inicialmente, a base de dados de *likes* está vazia;
- São efetuadas dois *likes* à planta X e cinco *likes* à planta Y através da aplicação móvel;
- Os dados são enviados da aplicação móvel para o servidor.
- O utilizador acede à aplicação de visualização de estatísticas e escolhe a opção de visualizar o número de likes associados às plantas.
- O utilizador verifica que são apresentadas dois likes para a planta X e cinco likes para a planta Y.

Cenário 3: Teste ao número de partilhas de *posts* na rede social Facebook associadas às plantas

- Inicialmente, a base de dados de partilhas no Facebook está vazia;
- São efetuadas duas partilhas para a planta X e cinco partilhas para a planta Y, na rede social Facebook, através da aplicação móvel;
- Os dados são enviados da aplicação móvel para o servidor.
- O utilizador acede à aplicação de visualização de estatísticas e escolhe a opção de visualizar o número de partilhas na rede social Facebook associadas às plantas.
- O utilizador verifica que são apresentadas duas partilhas para a planta X e cinco partilhas para a planta Y.

Cenário 4: Teste ao número de partilhas(twitts) na rede social Twitter associadas às plantas

- Inicialmente, a base de dados de partilhas no Twitter está vazia;
- São efetuadas duas partilhas para a planta X e cinco partilhas para a planta Y, na rede social Twitter, através da aplicação móvel;
- Os dados são enviados da aplicação móvel para o servidor.
- O utilizador acede à aplicação de visualização de estatísticas e escolhe a opção de visualizar o número de partilhas na rede social Twitter associadas às plantas.
- O utilizador verifica que são apresentadas duas partilhas para a planta X e cinco partilhas para a planta Y.

Cenário 5: Teste ao número de comentários associados às partilhas de dados na rede social Facebook.

- Inicialmente, a base de dados de partilhas no Facebook está vazia;
- São efetuadas duas partilhas para a planta X e uma partilha para a planta Y, na rede social Facebook, através da aplicação móvel;
- Os dados são enviados da aplicação móvel para o servidor.
- São adicionados 3 comentários em cada uma das duas partilhas da planta X e 2 comentários na partilha da planta Y.
- O utilizador acede à aplicação de visualização de estatísticas e escolhe a opção de visualizar o número de comentários associados aos *Posts* das plantas na rede social Facebook;
- O utilizador verifica que é apresentado um total de 6 comentários para a planta X e 2 comentários para a planta Y.

Cenário 6: Teste à filtragem do número de visitas às plantas num intervalo de tempo definido por horas

- Inicialmente, a base de dados do numero de visitas está vazia;
- São efetuadas 9 visitas às plantas entre as 10h30m e as 11h00 e 4 visitas entre as 14h00 e as 15h00;
- Os dados são envidados da aplicação móvel para o servidor;
- O utilizador acede à aplicação de visualização de estatísticas e escolhe a opção de visualizar o numero de visitas às plantas, aplicando um filtro por horas entre as 9h00 e as 11h30.
- A aplicação apresenta no total 9 visitas, correspondentes às visitas inseridas entre as 10h30 e as 11h00.

Cenário 7: Teste à filtragem do número de visitas às plantas num intervalo de tempo definido por dias

- Inicialmente, a base de dados do numero de visitas está vazia;
- São efetuadas 9 visitas no dia 11 de Junho, 4 visitas no dia 15 de Junho e 5 visitas no dia 22 de Junho.
- Os dados são envidados da aplicação móvel para o servidor;

- O utilizador acede à aplicação de visualização de estatísticas e escolhe a opção de visualizar o numero de visitas às plantas, aplicando um filtro por dias entre o dia 10 de Junho e o dia 18 de Junho;
- A aplicação apresenta no total 13 visitas, correspondentes às visitas inseridas para os dias 11 e 15 de Junho.

Cenário 8: Teste à filtragem do número de visitas às plantas num intervalo de tempo definido por dias e horas simultaneamente.

Cenário 9: Teste à filtragem do número de *likes* associados às plantas num intervalo de tempo definido por dias

Cenário idêntico ao cenário 7. No entanto, neste caso, são apresentados os resultados para o numero de *likes* associados às plantas.

Cenário 10: Teste à filtragem do número de likes associados às plantas num intervalo de tempo definido por horas

Cenário Semelhante ao cenário 6. No entanto, neste caso, são apresentados os resultados para o número de *likes* associados as plantas.

Cenário 11: Teste à filtragem do número de *likes* associados às plantas num intervalo de tempo definido por dias e horas simultaneamente

Cenário idêntico ao cenário 8. No entanto, neste caso, são apresentados os resultados para o numero de *likes* associados às plantas.

Cenário 12: Teste à filtragem do número de partilhas na rede social Facebook num intervalo de tempo definido por dias

Cenário idêntico ao cenário 7. No entanto, neste caso, são apresentados os resultados para as partilhas na rede social Facebook.

Cenário 13: Teste à filtragem do número de partilhas na rede social Facebook num intervalo de tempo definido por horas

Cenário Semelhante ao cenário 6. No entanto, neste caso, são apresentados os resultados para o número de partilhas na rede social Facebook associadas as plantas.

Cenário 14: Teste à filtragem do número de partilhas na rede social Facebook num intervalo de tempo definido por dias e horas

Cenário idêntico ao cenário 8. No entanto, neste caso, são apresentados os resultados para as partilhas na rede social Facebook.

Cenário 15: Teste à filtragem do número de partilhas(twitts) na rede social Twitter num intervalo de tempo definido por dias

Cenário idêntico ao Cenário 7. No entanto, neste caso, são apresentados os resultados para as partilhas efetuadas na rede social Twitter.

Cenário 16: Teste à filtragem do número de partilhas(twitts) na rede social Twitter num intervalo de tempo definido por horas

Cenário Semelhante ao cenário 6. No entanto, neste caso, são apresentados os resultados para o número de partilhas na rede social Twitter associadas as plantas.

Cenário 17: Teste à filtragem do número de partilhas(twitts) na rede social Twitter num intervalo de tempo definido por dias e horas

Cenário semelhante ao cenário 8. No entanto, neste caso, são apresentados os resultados para as partilhas efetuadas na rede social Twitter.

Cenário 18: Teste à filtragem do número de comentários presentes nos posts associados a uma planta, na rede social Facebook, num intervalo de tempo definido por dias

Cenário semelhante ao cenário 7. No entanto, neste caso, são apresentados os resultados para os comentários nos posts do Facebook associados às plantas.

Cenário 19: Teste à filtragem do número de comentários presentes nos posts associados a uma planta, na rede social Facebook, num intervalo de tempo definido por horas

Cenário Semelhante ao cenário 6. No entanto, neste caso, são apresentados os resultados para para os comentários nos posts do Facebook associados às plantas.

Cenário 20: Teste à filtragem do número de comentários presentes nos posts associados a uma planta, na rede social Facebook, num intervalo de tempo definido por dias e horas

Cenário semelhante ao cenário 8. No entanto, neste caso, são apresentados os resultados para os comentários nos posts do Facebook associados às plantas.

Como referido mais acima, os cenários apresentados foram repetidos alterando os valores dos filtros para a filtragem dos dados. Apenas são aqui apresentados os tipos de cenários criados para testar a aplicação.

Tal como para os testes da aplicação móvel em ambiente real, são efetuadas várias iterações dos testes até que todos os testes definidos sejam realizados e passados com sucesso. Apresentam-se a seguir, na tabela 26, os resultados obtidos para cada uma das iterações.

Iteração 1

Cenário	Reultado	Observações
Cenário 1	Passou	
Cenário 2	Passou	
Cenário 3	Passou	
Cenário 4	Passou	

Cenário 5	Passou	
Cenário 6	Falhou	
Cenário 7	Passou	
Cenário 8	Falhou	A filtragem não estava a considerar corretamente o intervalo de tempo em horas
Cenário 9	Passou	
Cenário 10	Passou	
Cenário 11	Falhou	A filtragem não estava a considerar corretamente o intervalo de tempo em horas
Cenário 12	Passou	
Cenário 13	Passou	
Cenário 14	Falhou	A filtragem não estava a considerar corretamente o intervalo de tempo em horas
Cenário 15	Passou	
Cenário 16	Passou	
Cenário 17	Falhou	A filtragem não estava a considerar corretamente o intervalo de tempo em horas
Cenário 18	Passou	
Cenário 19	Passou	
Cenário 20	Falhou	A filtragem não estava a considerar corretamente o intervalo de tempo em horas

Tabela 26: Resultados obtidos para a primeira iteração de testes da componente de visualização de estatísticas

Concluída a primeira iteração de testes, verificou-se a existência de falhas na aplicação, verificando-se a falha de alguns testes executados. As falhas detectadas na primeira iteração ocorriam em testes relacionados com a filtragem de dados em intervalos de tempo que combinam dias e horas.

Procedeu-se às alterações necessárias no código para corrigir as falhas encontradas, de modo a proceder a uma segunda iteração de testes. São apresentados na tabela 27 os resultados obtidos para a segunda iteração de testes.

Iteração 2

Cenário	Resultado	Observações
Cenário 1	Passou	
Cenário 2	Passou	
Cenário 3	Passou	
Cenário 4	Passou	
Cenário 5	Passou	
Cenário 6	Passou	

Cenário 7	Passou	
Cenário 8	Passou	
Cenário 9	Passou	
Cenário 10	Passou	
Cenário 11	Passou	
Cenário 12	Passou	
Cenário 13	Passou	
Cenário 14	Passou	
Cenário 15	Passou	
Cenário 16	Passou	
Cenário 17	Passou	
Cenário 18	Passou	
Cenário 19	Passou	
Cenário 20	Passou	

Tabela 27: resultados obtidos para a segunda iteração de testes da componente de visualização de estatísticas

Após ter realizado a segunda iteração de testes, verificou-se que todos os testes definidos deram resultados positivos, não falhando nenhum.

Compatibilidade com browsers

A aplicação foi testada em vários browsers, de modo a verificar a compatibilidade da biblioteca Polymaps com os diferentes browsers.

Internet Explorer: A biblioteca Polymaps só funciona com este browser à partir da versão 9 do Internet Explorer. Versões mais antigas deste browser não conseguem carregar os elementos que pertencem à biblioteca Polymaps.

Mozilla Firefox: A utilização do Polymaps com este browser apresenta algumas limitações. Os elementos conseguem ser carregados e manipulados, no entanto, o mapa

Google Chrome: Todas as funcionalidades da aplicação funcionam bem recorrendo a este browser;

Safari: Todas as funcionalidades da aplicação funcionam bem recorrendo a este browser.

Opera: A utilização do Polymaps com este browser apresenta as limitações enunciadas para a utilização do browser Firefox.

Capítulo 8

Conclusão e Trabalho Futuro

O presente documento teve como objetivo apresentar o desenvolvimento de um guia turístico móvel para o Jardim Botânico de Coimbra. O trabalho dividiu-se em duas fases: A primeira fase foi dedicada ao estudo de soluções do mesmo ramo existentes no mercado, assim como ao estudo de soluções tecnológicas essenciais para proceder ao desenvolvimento da aplicação. Também foram levantados os requisitos da aplicação assim como definida a arquitetura da mesma; A segunda fase foi dedicada ao desenvolvimento das componentes assim como à fase de testes sobre as mesmas e correções de erros detetados.

A passagem pelas várias fases do projeto permitiram no final chegar ao produto desejado. Através do estudo do Estado da Arte, feito no início do projeto, foi possível analisar soluções existentes no mercado para guias turísticos móveis. Foram analisadas algumas soluções consideradas como sendo as mais populares encontradas na literatura, bem como algumas soluções consideradas relevantes tendo em conta o produto final a desenvolver. Para além disso, foram analisadas e comparadas várias soluções tecnológicas que poderiam ser utilizadas nas aplicações desenvolvidas.

A análise dos requisitos permitiu definir com facilidade as funcionalidades a serem implementadas nas aplicações desenvolvidas. Procedeu-se à definição da arquitetura do sistema, identificando elementos relevantes para a mesma como por exemplo o modo de comunicação entre o servidor e a aplicação móvel e aplicação de visualização de estatísticas. Mecanismos de sincronização de dados também foram identificados.

Após as fases de especificação de requisitos e arquitetura foi iniciada a fase de implementação das várias componentes que compõem o projeto. Foi primeiro criada a componente servidor, utilizando a API existente da empresa e adicionando os modelos necessários. Após isto, o autor participou primeiro no desenvolvimento de uma outra aplicação móvel. Após isto, iniciou-se o desenvolvimento da aplicação móvel para o Jardim Botânico. Considera-se que se chegou a um produto que consegue melhorar a experiência do visitante, principalmente através da componente de realidade aumentada, oferecendo uma alternativa à visualização de informação em duas dimensões. Finalmente, desenvolveu-se a aplicação para visualização de estatísticas. De forma a validar as componentes desenvolvidas procedeu-se à realização de testes. Para a API REST foram utilizados testes automatizados através de ferramentas do Django e Tastypie. Para a aplicação móvel, foram realizados vários tipos de testes. Foram realizados testes de usabilidade que permitiram concluir que a aplicação parece apresentar bons índices de usabilidade. Foram realizados testes em ambiente real (no interior do Jardim Botânico) para verificar se a aplicação cumpria os requisitos definidos inicialmente. Também foram realizados testes para verificar se a sincronização dos dados entre a aplicação móvel e o

servidor ocorria com sucesso. Finalmente, também foram realizados testes para verificar o funcionamento da aplicação de visualização de estatísticas. Os testes foram realizados em iterações até passarem todos. Para a componente servidor foi possível passar em todos os testes definidos. Para a aplicação móvel, foram necessárias duas iterações até conseguir passar em todos os testes em ambiente real. Testes de usabilidade foram realizados e apresentaram um resultado final de 80 (numa escala de 0 a 100). Quanto à aplicação de visualização de estatísticas, após duas iterações de testes todos eles foram capazes de passar com sucesso.

Análise crítica

O autor considera que o resultado final correspondeu às expectativas iniciais. Para além do que estava definido, foi possível participar no desenvolvimento de uma outra aplicação móvel orientada para o turismo. Foi possível criar a componente de realidade aumentada que tornou a aplicação bastante mais interessante e atrativa. Para além de um guia geral que estava inicialmente definido, a utilização da realidade aumentada foi estendida ao módulo das rotas, permitindo ao utilizador seguir uma rota através da realidade aumentada. A procura por uma determinada planta também é possível através da realidade aumentada. Torna-se de facto uma alternativa à visualização dos dados em duas dimensões. A integração das redes sociais na aplicação móvel também foi conseguida. De facto, as componentes consideradas diferenciadoras para a aplicação móvel (realidade aumentada e redes sociais) foram implementadas. Todavia, existe um aspeto que poderia ter sido melhorado, uma vez que ficou uma métrica por representar na aplicação de visualização de estatísticas. O tempo gasto no desenvolvimento da outra aplicação móvel pode ser considerado um motivo para a falta de tempo necessária ao desenvolvimento da última métrica.

Principais obstáculos

O principal obstáculo à realização da aplicação móvel foi a incapacidade da entidade gestora do Jardim Botânico da Universidade de Coimbra em fornecer conteúdos para introduzir na aplicação. Não foram fornecidos quaisquer conteúdos sobre as plantas presentes no seu interior. O autor teve que proceder à sua própria recolha de dados, não oferecendo tanta qualidade quanto aos dados apresentados na aplicação. Com a colaboração da entidade gestora do Jardim Botânico, acredita-se que a aplicação poderia fornecer informação de maior qualidade e mais diversificada.

Trabalho futuro

Após a conclusão deste trabalho, o autor considera que existem tópicos que podem ser explorados na aplicação móvel e na aplicação de visualização de estatísticas.

- Na aplicação móvel, a parte da realidade aumentada pode ser explorada, para permitir a visualização de elementos em 3D. Com dados oficiais, fornecidos pela entidade gestora do Jardim Botânico é possível tornar a aplicação mais rica em termos da qualidade, quantidade e variação do tipo de informação apresentada. O leitor de códigos QR também pode ser implementado futuramente, caso a entidade gestora do Jardim Botânico decida por exemplo instalar placas com códigos QR ao lado de cada planta. Outra melhoria que pode ser feita diz respeito ao mapa

utilizada na aplicação. Foi utilizada a biblioteca OpenStreetMaps para permitir à aplicação utilizar o mapa sem requerer ligação à internet. No entanto, o nível máximo de zoom apresentado é limitado. Utilizar um mapa recorrendo à utilização da rede móvel poderia resolver este problema.

- Quanto a componente de visualização de estatísticas, esta pode ser explorada para permitir por exemplo exportar os resultados sobre a forma de gráficos de modo a poder visualizar mais facilmente comparações de resultados em intervalos de tempos diferentes por exemplo.

Bibliografia

- [1] Wieland Schwinger, Christoph Grun, Brigit Proll, Werner Retchitzegger, A. Schauerhuber, Context-awareness in Mobile Tourism Guides, *Handbook of Research on Mobile Multimedia, Second Edition*, 2005, pp 534-552
- [2] Jorg Baus, Keith Cheverst, Christian Kray, A Survey of Map-based Mobile Guides. *Map-based Mobile Services*, 2005, pp. 193-209
- [3] Grogory D. Abowd, Christopher G. Atkeson, Json Hong, Sue Long, Rob Kooper, Mike Pinkerton, Cyberguide: A Mobile context-aware tour guide. *Wireless Networks 3*, 1997 , pp. 421-423
- [4] Tong Chan, Analysis of Critical Success Factor of Mobile Location-based Services (Tese de Mestrado), *Royal Institute of Technology*, 2009
- [5] Natalie Jun Pei Chin, Critical Success Factors os Location-Based Services (Tese de Mestrado), *University of Nebraska-Lincoln*, 2012
- [6] Antonio Grillo, Alessandro Lentini, Marco Querini, Giuseppe F. Italiano, High Capacity Colored Two Bidimensional Codes. *Proceedings of the International Multiconference on Computer Science and Information Technology*, 2010, 709-716
- [7] LaunchMedia, 2D Barcode Implementation At National Retail White Paper, 2009.
- [8] Mark van Seten, Stanislav Pokraev, Johan Koolwaaij, Context Aware Recommendations in the Mobile Tourist Application Compass. *Adaptive Hypermedia and Adaptive Web-Based Systems*, 2004, pp. 235-244
- [9] Carlos Oliveira, Imperceptible Markers for Indoor Positioning and Identification on Mobile Travel Guides (Tese de Mestrado), Faculdade de Ciências e Tecnologias da Universidade de Coimbra, 2011
- [10] Tek Yong Lim, Designing the Next Generation of Mobile Tourism Application based on Situation Awareness. *Southeast Asian Network of Ergonomics Societies Conferense (SEANES)*, 2012
- [11] Jin Li, Qinmin Wu, Xucqian Li, Daoli Zhu, Context-based Personalized Mobile Tourist Guide, *IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009 (ICIS 2009)*, pp. 607-611
- [12] Security and Privacy Issues in Location Based Service:, Disponível online em <http://www.roseindia.net/technology/LBS/security-and-privacy-issues-lbs.shtml>, Februray, 8 2008 (acedido pela última vez no dia 01 de Julho de 2013)
- [13] The Top Seven Alternatives to the Google Maps API, Disponível online em <http://www.netmagazine.com/features/top-seven-alternatives-google-maps-api> (acedido pela última vez no dia 23 de Janeiro de 2013)

- [14] OenStreetMap Wiki, Disponível online em http://wiki.openstreetmap.org/wiki/Pt-br:Press_Kit (acedido pela última vez no dia 01 de Julho de 2013)
- [15] GeoTools, Disponível online em <http://www.geotools.org/about.html> (acedido pela última vez no dia 01 de Julho de 2013)
- [16] OpenLayers: Free Maps for the Web, Disponível online em <http://openlayers.org/> (acedido pela última vez no dia 01 de Julho de 2013)
- [17] PolyMaps: A Javascript Library for image – and vector–tiled maps Usin SVG, Disponível online em <http://polymaps.org/> (acedido pela última vez no dia 01 de Julho de 2013)
- [18] Paulo Pombinho de Matos, Maria Beatriz Carmos, Ana Paula Afonso, Visualização de Informação Georeferenciada em Dispositivos Móveis, 2007
- [19] Judy M. Olson, Using Colors in Statistical Graphs and Maps, 2002
- [20] Orientação metodológica para construção e leitura de mapas temáticos, Disponível online em <http://confins.revues.org/3483?id=3483> (acedido pela última vez no dia 01 de Julho de 2013)
- [21] Carlos Manuel Azevedo de Jesus, Serviços móveis baseados na localização com realidade aumentada (Tese de Mestrado), Universidade de Aveiro – Departamento de Comunicação e Arte, 2009
- [22] Anders Henrysson, Bringing Augmented Reality to Mobiles Phones (Tese de Mestrado), Department of Science and Technology Linkopings universitet, 2007
- [24] Carlos Jesus, Lídia J. Oliveira L. Silva, Potencialidade dos serviços móveis de Realidade Aumentada aplicados ao Turismo, 2009 ,
http://www.ic.ufmt.br:8080/c/document_library/get_file?p_l_id=58070&folderId=59704&name=DLFE-2175.pdf
- [25] Vuforia developer, Disponível online em <https://developer.vuforia.com/resources/dev-guide/getting-started> (acedido pela última vez no dia 1 de Julho de 2013)
- [26] Junaio, Developer Documentation AREL API, Disponível online em <http://www.junaio.com/develop/docs/> (acedido pela última vez no dia 1 de Julho de 2013)
- [27] Wikitude, The App Developer Agency Showcases, Disponível online em <http://www.wikitude.com/developer/sdk> (acedido pela última vez no dia 1 de Julho de 2013)
- [29] Django: The web framework for perfectionists with deadlines, hDisponível online em <https://www.djangoproject.com/> (acedido pela última vez no dia 1 de Julho de 2013)

[30] Tastypie, Disponível online em <http://tastypieapi.org/> (acedido pela última vez no dia 1 de Julho de 2013)

[31] PostGis, Disponível online em <http://postgis.refractor.net/> (acedido pela última vez no dia 1 de Julho de 2013)

[32] Why We Use PostgreSQL for Web Application Development, Disponível online em http://support.summersault.com/why_postgresql.html (acedido pela última vez no dia 1 de Julho de 2013)

[33] Oracle vs PostgreSQL, Disponível online em <http://www.scriptol.com/software/oracle.php> (acedido pela última vez no dia 1 de Julho de 2013)

[34] Django, Django Documentation, FAQ: General, Disponível online em <https://docs.djangoproject.com/en/dev/faq/general/#faq-mtv> (acedido pela última vez no dia 1 de Julho de 2013)

[35] Getting Started: The Graph API, Disponível online em <https://developers.facebook.com/docs/getting-started/graphapi/> (acedido pela última vez no dia 1 de Julho de 2013)

[36] Hermann Anegg, Harald Kunczier, Elke Michlmayer, Gunther Pospischil, Martina Umlauf, LoL@: Designing a location based UMTS application. *e&i Elektrotechnik und Informationstechnik, Volume 119, issue 2, 2002*, pp. 48-51

[37] Alexandre Lopes, InovWine Desenvolvimento de Aplicação Móvel de Apoio à Agricultura de Precisão (Tese de Mestrado), Faculdade de Ciências e Tecnologias da Universidade de Coimbra, 2011.

[38] Jochen Schiller, Agnès Voisard, *Location-Based Services*, pp. 14-20

[39] Application: Mobile Computing Systems, Disponível online em <http://www.agentfactory.com/~rem/PRISM-InfoPack/Mobile/index.html> (acedido pela última vez no dia 1 de Julho de 2013)

[40] Luís Carlos Santos Da Conceição, Estudos de sistemas de realidade aumentada aplicados ao turismo (Tese de Mestrado), 2006

[41] Propostas de estágio 2012/2013, Disponível online em <https://estagios.dei.uc.pt/cursos/mei/2012-2013/propostas-de-estagio-2012-2013/?id=1014> (acedido pela última vez no dia 1 de Julho de 2013)

[42] Realidade Aumentada, Disponível online em http://www.marcaspatentes.pt/files/collections/pt_PT/1/300/301/Realidade%20Aumentada.pdf

[43] iTunes Preview Regensburg, Disponível online em <https://itunes.apple.com/us/app/regensburg/id360983806> (acedido pela última vez no dia 1 de Julho de 2013)

[44] What are the Different Types of 2D Barcodes?, Disponível online em <http://www.sumlung.com/en/about-barcode/310.html> (acedido pela última vez no dia 1 de Julho de 2013)

[45] Can I use... , Disponível online em <http://caniuse.com/#agents=mobile&cats=SVG> (acedido pela última vez no dia 1 de Julho de 2013)

[46] Go offline with Google Maps for Android, Disponível online em <http://google-latlong.blogspot.pt/2012/06/go-offline-with-google-maps-for-android.html> (acedido pela última vez no dia 1 de Julho de 2013)

[47] Google Maps, Disponível online em http://en.wikipedia.org/wiki/Google_Maps (acedido pela última vez no dia 1 de Julho de 2013)

[48] Chapter 26. Datamatrix (2D-Barcode), Disponível online em <http://jpgraph.net/download/manuals/chunkhtml/ch26.html> (acedido pela última vez no dia 1 de Julho de 2013)

[49] Qr Code X Datamatrix X Microsoft Tag X Beetag, Disponível online em <http://www.domicioneto.com/marketing-digital/qr-code-datamatrix-microsoft-tag-beetag/> (acedido pela última vez no dia 1 de Julho de 2013)

[50] IDC Worldwide Mobile Phone Tracker, Disponível online em <http://www.idc.com/getdoc.jsp?containerId=prUS23946013> (acedido pela última vez no dia 1 de Julho de 2013)

[51] IDC Worldwide Mobile Phone Tracker, Disponível online em <http://www.idc.com/getdoc.jsp?containerId=prUS23523812> (acedido pela última vez no dia 1 de Julho de 2013)

[52] MyOrpheo, Disponível online em <http://myorpheo.com/gardens-of-international-perfume-museum/> (acedido pela última vez no dia 1 de Julho de 2013)

[53] Getting Started: The Graph API, Disponível online em <https://developers.facebook.com/docs/getting-started/graphapi/> (acedido pela última vez no dia 1 de Julho de 2013)

[54] Introduction to User Stories, Disponível online em <http://www.agilemodeling.com/artifacts/userStory.htm> (acedido pela última vez no dia 1 de Julho de 2013)

[55] System Use Cases, Disponível online em <http://www.agilemodeling.com/artifacts/systemUseCase.htm> (acedido pela última vez no dia 1 de Julho de 2013)

[56] PDF417 Specification, Disponível online em <http://www.morovia.com/education/symbology/pdf417.asp> (acedido pela última vez no dia 1 de Julho de 2013)

[57] Django Documentation, Disponível online em <https://docs.djangoproject.com/en/dev/faq/general/#faq-mtv> (acedido pela última vez no dia 1 de Julho de 2013)

[58] Códigos Bidimensionais | 2D, Disponível online em <http://www.oxxcode.com.br/codigo-bidimensional-2d/> (acedido pela última vez no dia 1 de Julho de 2013)

[59] User authentication in Django, Disponível online em <https://docs.djangoproject.com/en/dev/topics/auth/> (acedido pela última vez no dia 1 de Julho de 2013)

[60] Using the Twitter Search API, Disponível online em <https://dev.twitter.com/docs/using-search> (acedido pela última vez no dia 1 de Julho de 2013)

[61] America's National Parks, Disponível online em <https://dev.twitter.com/docs/using-search> (acedido pela última vez no dia 1 de Julho de 2013)

[62] WALKME – À descoberta das maravilhas da Ilha da Madeira, Disponível online em <http://pplware.sapo.pt/smartphones/android/walkme-a-descoberta-das-maravilhas-naturais-da-ilha-da-madeira/> (acedido pela última vez no dia 1 de Julho de 2013)

[63] JSON: The Fat-Free Alternative to XML, Disponível online em <http://www.json.org/xml.html> (acedido pela última vez no dia 1 de Julho de 2013)

[64] Google Social Graph API, Disponível online em <https://developers.google.com/social-graph/> (acedido pela última vez no dia 1 de Julho de 2013)

[65] Barcodes Help, Disponível online em <http://www.dlsoft.com/dlHelps/helps/barcodes/default.htm#!maxicode2.htm> (acedido pela última vez no dia 1 de Julho de 2013)

[66] Android Developers – The AndroidManifest.xml File, Disponível online em <http://developer.android.com/guide/topics/manifest/manifest-intro.html> (acedido pela última vez no dia 01 de Julho de 2013)

[67] Android Developers – Swipe Views, Disponível online em <http://developer.android.com/design/patterns/swipe-views.html> (acedido pela última vez no dia 01 de Julho de 2013)

[68] Measuring Usability with the System Usability Scale (SUS), Disponível online em <http://www.measuringusability.com/sus.php> (acedido pela última vez no dia 1 de Julho de 2013)

[69] Daniel Silva, Vasco Moreira, “An Interactive Augmented Reality Battleship Game Implementation”, Proceedings of Learning with Games 2007, September 24--26 2007, Sophia Antipolis, France (Marco Taisch and Jacopo Cassina eds.), pp. 213-219, 2007

[70] Pedro Abreu, Pedro Mendes. "Mastermind: an Augment Reality Approach", Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts, pp.205-210, 2008

[71] Google, Maps for Mobile, Disponível online em https://support.google.com/gmm/answer/2650380?hl=en&ref_topic=2649131 (acedido pela última vez no dia 01 de Julho de 2013)

Anexos

Anexo I - Códigos Bidimensionais

Códigos bidimensionais são códigos de barras bidimensionais que servem para armazenar informações. Estes códigos permitem automatizar processos de trabalho sem a intervenção do ser humano. São códigos amplamente utilizados uma vez que são rápidos e precisos. Os principais desafios dos códigos bidimensionais residem na sua necessidade de armazenarem mais informação e mais tipos de dados, sem comprometer a sua eficiência prática [6]. Códigos bidimensionais tornaram-se muito populares, devido principalmente à sua velocidade de leitura, precisão e características funcionais. De modo a aumentar a capacidade de dados armazenada, códigos bidimensionais introduzem a capacidade de guardar a informação em duas direções ou seja, nas direções verticais e horizontais.

Códigos bidimensionais podem ter aplicabilidade nas seguintes situações [58]:

- Industrias: Utilizados em logística, notas fiscais, rastreamento de produtos, identificação de peças;
- Comércio: Controlo de mercadorias, stocks, clientes;
- Bancos: Identificação de cheques, antifraudes em pagamentos;
- Seguradoras: Controlo de documentos, registo de seguros;
- Hospitais: Fichas médicas, credenciais, controlo de materiais, exames, controlo de medicamento.

Principais soluções analisadas

Existem muitos tipos de códigos bidimensionais convencionais. A principal diferença entre todos eles encontra-se na quantidade de dados que conseguem armazenar. Outro aspeto importante que permite diferenciar os códigos bidimensionais é a capacidade de correção dos erros que cada código apresenta. Segue-se uma descrição dos códigos bidimensionais mais populares presentes no mercado [44]. Estas soluções foram analisadas tendo em vista a escolha de um tipo de código bidimensional para utilizar na aplicação a desenvolver.

Códigos QR: Este tipo de código é *open source*, ou seja, pode ser criado por qualquer entidade e lido a partir de um leitor qualquer. Em termos de leitura, é considerado como sendo o código mais tolerante, uma vez que permite a leitura com vários ângulos e várias condições de iluminação. Este tipo de código pode conter qualquer tipo de dados, tendo capacidade máxima de 7089 caracteres numéricos, 4296 caracteres alfanuméricos, 2953 bytes de dados binários. Apresenta uma taxa de correção de erros, dividida em vários níveis, L, M, Q e H, que corrigem respetivamente 7%, 15%, 25% e 30% de erros. Outro aspeto relevante é que o tamanho é variável, existindo os formatos versão 1 (21x21) e versão 40 (177x177). [7]



Figura 40: Representação de um código QR [44]

Microsoft Tag: Trata-se de um tipo de código bidimensional proprietário da companhia Microsoft. Este tipo de código apenas pode ser criado e lido a partir de uma plataforma Microsoft. Esta tecnologia destaca-se em relação as outras, por ser colorida. Consiste em linhas de *strings* e símbolos (triângulos) com quatro tipos de cores: preto, vermelho, amarelo e verde. A leitura e decodificação deste tipo de TAG requer ligação à internet, utilizando obrigatoriamente aplicações e servidores da Microsoft. Este tipo de código tem uma capacidade de armazenamento de 3500 caracteres alfanuméricos [7] [9].

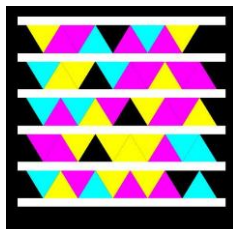


Figura 41: Representação de um código Microsoft Tag [49]

DataMatrix Code: Muito similar a códigos QR. No entanto foi criado para utilizar resoluções extremamente baixas, tendo uma capacidade de armazenamento mais baixa do que um QR Code mas maior do que um código PDF417. Este tipo de código consiste num conjunto de quadrados pretos e brancos a formar um padrão quadrado ou retangular. [49] [48]. A capacidade máxima de armazenamento é de 2335 caracteres alfanuméricos ou 3116 caracteres numéricos. O tamanho do símbolo pode ir de 10x10 até um máximo de 144x144. Este tipo de código usa o método *Reed-Solomon* para correção de erros, tendo uma capacidade máxima de correção de erros de 30% [7] [9].



Figura 42: Representação de um código DataMatrix [49]

PDF417: Um código deste tipo consiste num conjunto de 3 a 90 linhas. Contém entre 2 e 512 caracteres para correção de erros, divididos em 8 níveis de correção de erros. Consegue codificar um máximo de 1850 caracteres alfanuméricos, 2710 dígitos ou 1108 bytes. Cada uma das linhas do código está dividida em 7 zonas: uma *quiet zone*, um padrão

de início, um símbolo indicador, 1 a 30 caracteres, outro símbolo indicador, um padrão de fim e outra *quiet zone* [56]. A capacidade de correção de erros depende do nível de segurança aplicado ao código. Existem 9 níveis de segurança. No nível 0 conseguem ser corrigidas *codewords* enquanto que no nível 8 conseguem ser corrigidas 512 *codewords*.

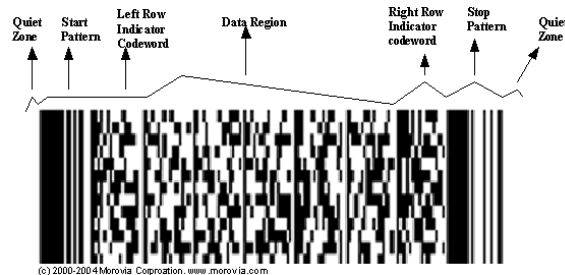


Figura 43: Representação de um código PDF417 [56]

Maxi Code: Trata-se de um código bidimensional desenvolvido pela *United Parcel Services* em 1992. No entanto, em vez de ser constituído por uma série de pontos quadrados, MaxiCode é composto por um *array* de uma polegada por uma polegada com 866 hexágonos interligados. Isto permite que o código seja pelo menos 15 por cento mais denso do que um código de ponto quadrado. Podem ser armazenados aproximadamente 100 caracteres ASCII em cada símbolo de 1 polegada quadrada. O símbolo pode ainda ser lido mesmo quando 25 por cento do símbolo foi destruído [65].

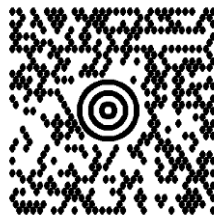


Figura 44: Representação de um código MaxiCode

ScanLife Codes: Este tipo de código permite eliminar a restrição da densidade do código, aliada a todos os outros tipos de código vistos até agora, sendo esta a principal características deste tipo de código bidimensional. [7]



Figura 45: Representação de um código ScanLife [7]

A Tabela 28 resume os diferentes tipos de códigos bidimensionais analisados, tendo em conta algumas características importantes: a quantidade de dados codificados, a capacidade de correção de erros, a capacidade de funcionar em modo *offline* e o tipo de especificação do código.

Código	Dados Codificados	Especificação aberta	Modo off.	Correção de Erros
QR Code	Máximo de 4296 caracteres alfanuméricos	Sim	Sim	Até 30 %
Microsoft TAG	Máximo de 3500 caracteres alfanuméricos	Não	Não	Não especificado
Data Matrix Code	Máximo de 2355 caracteres alfanuméricos	Sim	Sim	Até 30%
PDF417	Máximo de 1850 caracteres alfanuméricos	Sim	Sim	Corrige de 2 a 512 <i>codewords</i> dependendo do nível de segurança
Maxi Code	Máximo de 93 caracteres alfanuméricos	Sim	Sim	Até 25%
ScanLife Code	Não definido	Não	Não	Apenas 4%

Tabela 28: Revisão das tecnologias existentes para códigos bidimensionais

Anexo II - Prototipagem da aplicação móvel

Apresenta-se nesta secção a prototipagem da aplicação definida através da criação de *mockups*. São apresentados os *mockups* criados inicialmente pelo autor assim como os *mockups* criados pelo *designer* da empresa, fara efeito comparativo.

Menu principal da aplicação

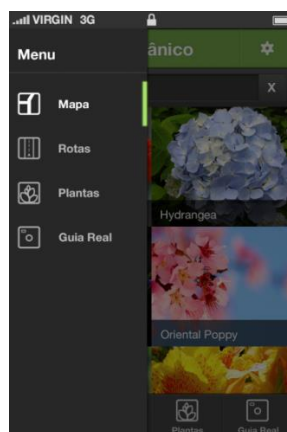


Figura 46: Menu principal da aplicação

A Figura 46 mostra o *mockup* para o menu principal da aplicação, onde estão presentes 4 opções seleccionáveis: Mapa (para visualizar o mapa do Jardim com as plantas), Rotas (Para aceder ao módulo de rotas), Plantas (para aceder à listagem das plantas existentes) e Guia Real (para aceder ao guia com realidade aumentada).

São apresentados na Figura 47 os *mockups* definidos pelo autor para o menu principal.

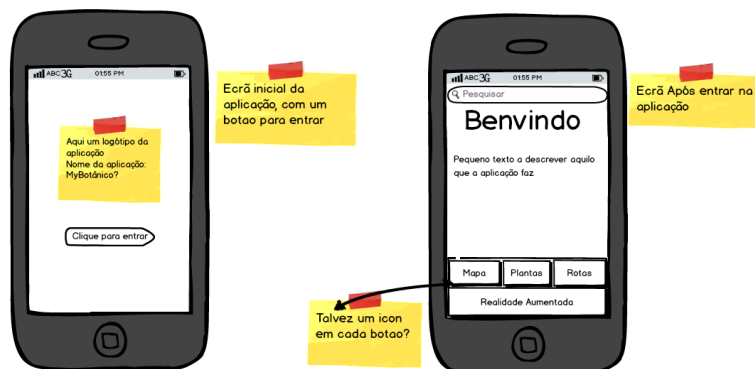


Figura 47: *Mockup* criado pelo autor para o Menu Principal

Listagem das plantas, filtros e BI da planta

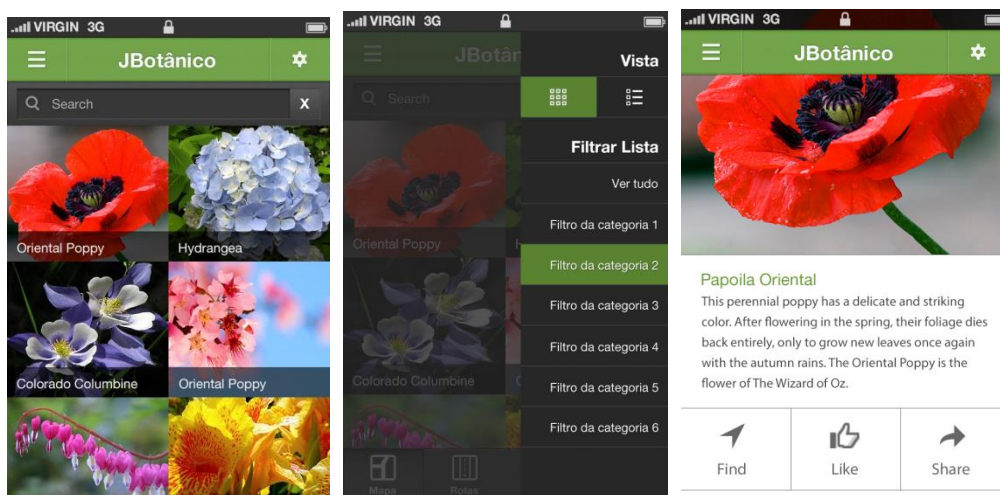


Figura 48: ecrãs da listagem das plantas(esquerda), filtros da listagem(centro) e BI da planta(direita)

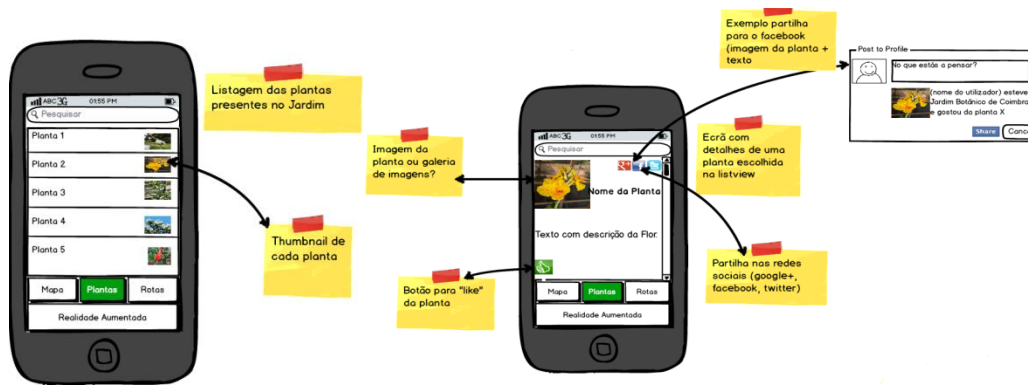


Figura 49: Mockups criados pelo autor para a listagem das plantas e BI das mesmas

A Figura 48 mostra os *mockups* definidos para a listagem das plantas (figura mais à esquerda), os filtros para a listagem das plantas (figura do centro) e o ecrã para apresentar os detalhes de uma planta (figura mais à direita). A figura 49 apresenta os *mockups* criados pelo autor.

Prototipagem do mapa da aplicação



Figura 50: prototipagem do mapa da aplicação

A Figura 50 mostra o *mockup* definido para a implementação do mapa. Utilizando o mapa, é possível visualizar as plantas existentes apresentando o nome e uma imagem da mesma. A figura 51 apresenta o *mockup* criado pelo autor.

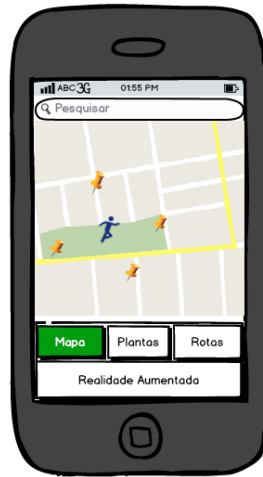


Figura 51: *Mockup* criado pelo autor para a utilização do mapa

Prototipagem do módulo de rotas da aplicação móvel

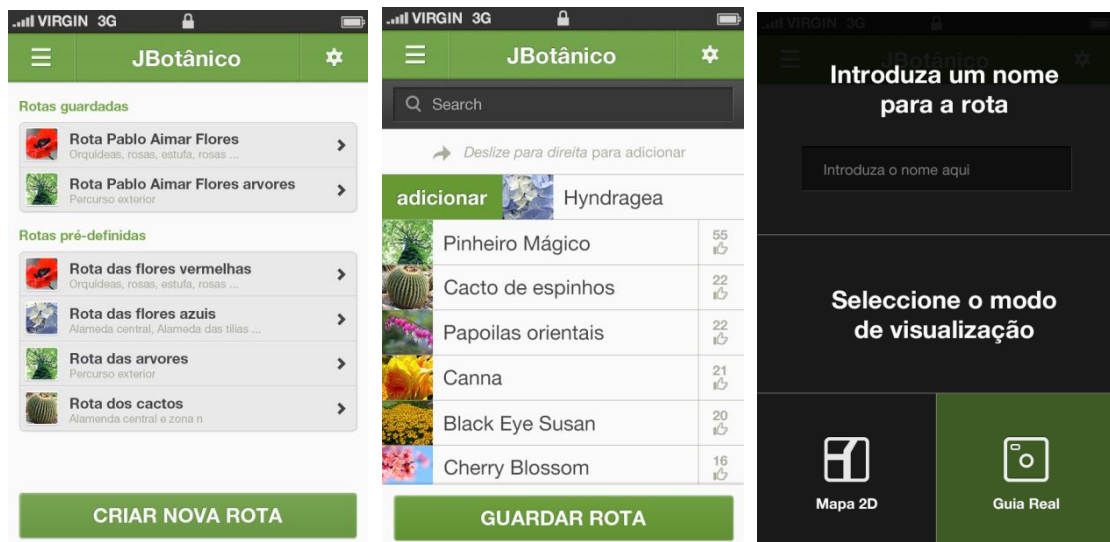


Figura 52: prototipagem da componente de rotas da aplicação

A Figura 52 mostra os *mockups* definidos para a criação do módulo de rotas. A figura mais à esquerda apresenta o ecrã pensado para a listagem das rotas. É feita a diferença entre as rotas pré-definidas e as rotas criadas pelo utilizador. A figura do meio mostra o ecrã pensado para a possibilidade do utilizador poder criar a sua própria rota, escolhendo as plantas que deseja visitar. A figura mais à direita apresenta o ecrã pensado para a escolha do modo de visualização da rota, uma vez que a rota pode ser seguida no mapa 2D ou através do guia real (realidade aumentada). A figura 51 apresenta os *mockups* criados pelo autor.

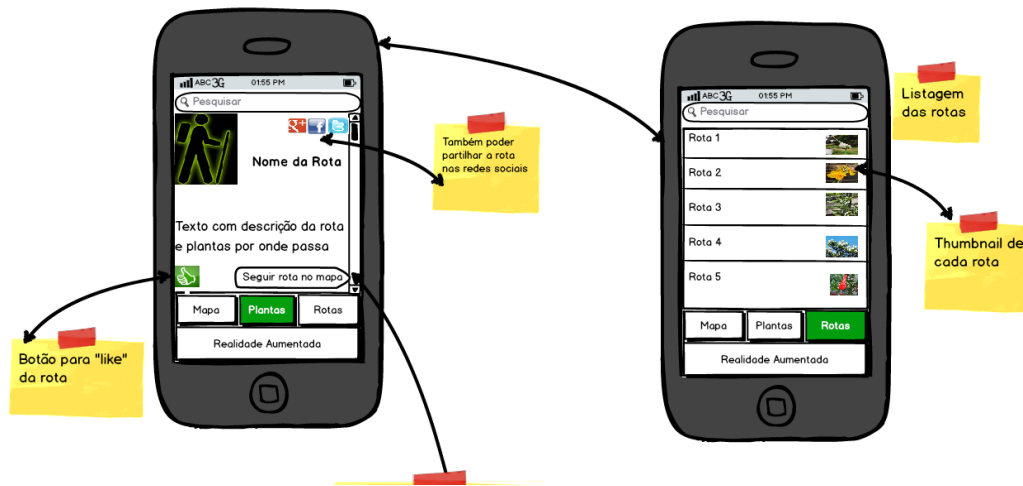


Figura 53: *Mockups* criados pelo autor para a componente de rotas

Prototipagem do módulo de realidade aumentada



Figura 54: prototipagem do guia real(realidade aumentada) da aplicação

A Figura 54 mostra o *mockup* definido para a utilização do Guia Real. No caso da sua utilização, para seguir uma rota, a aplicação apresenta indicações sobre a próxima planta a visitar assim como a distância até à mesma. A figura 55 apresenta o *mockup* criado pelo autor.



Figura 55: *Mockup* criado pelo autor para a componente de realidade aumentada

Anexo III - Ecrãs da aplicação móvel final

Apresentam-se nesta secção os ecrãs da aplicação final desenvolvida, assim como uma legenda para explicar a correspondência de cada um dos ecrãs.

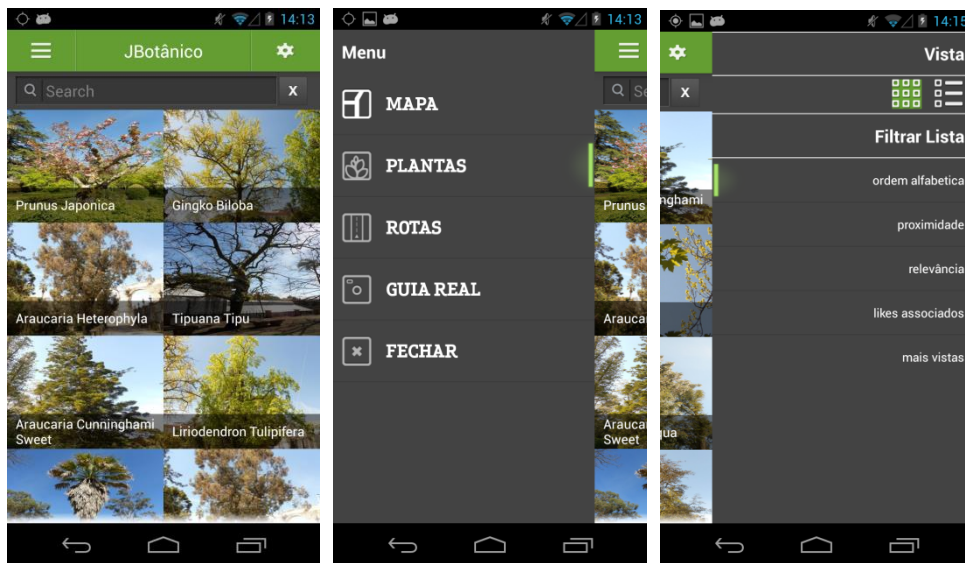


Figura 56: Representação do menu principal da aplicação (figura do meio), da listagem das plantas (figura da esquerda) e do menu de filtros à listagem das plantas (figura da direita)

A Figura 56 mostra o resultado final para a listagem das plantas (imagem mais à esquerda), para o menu principal (imagem ao centro) assim como para o menu dos filtros que podem ser aplicados à listagem das plantas (imagem mais à direita). Na imagem mais à direita, pode-se observar que é possível escolher o modo de visualização da listagem das plantas (por grelha ou por lista). Observa-se também a possibilidade de filtrar a lista por ordem alfabética, por proximidade, por relevância, em função dos *likes* associados a cada planta ou então em função do número de visitas por planta.

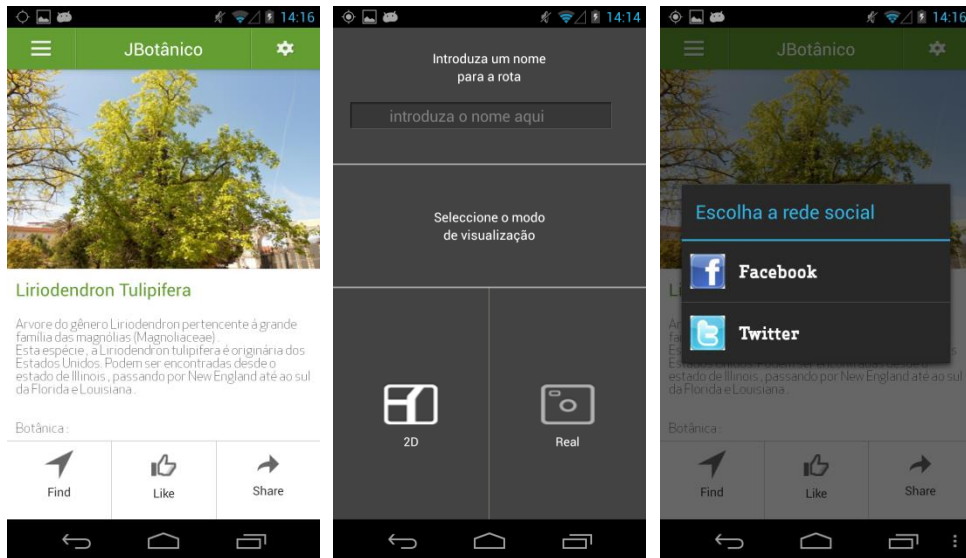


Figura 57: Representação do BI de uma planta (figura da esquerda), escolha do modo de visualização (figura do meio) e escolha da rede social para partilhar (figura da direita)

A Figura 57 mostra o resultado final para a apresentação da informação detalhada de uma planta (imagem mais à esquerda), para a escolha do modo de visualização de dados (2D ou guia real) assim como para a escolha da rede social para partilhar dados (imagem mais a direita). No ecrã de apresentação de detalhes da planta, observa-se na parte de baixo do ecrã a possibilidade de escolher por procurar a localização da planta no Jardim (Find) através do mapa 2D ou da realidade aumentada, escolher a opção de fazer “like” à planta (Like) assim como a opção de partilhar dados nas redes sociais (Share).

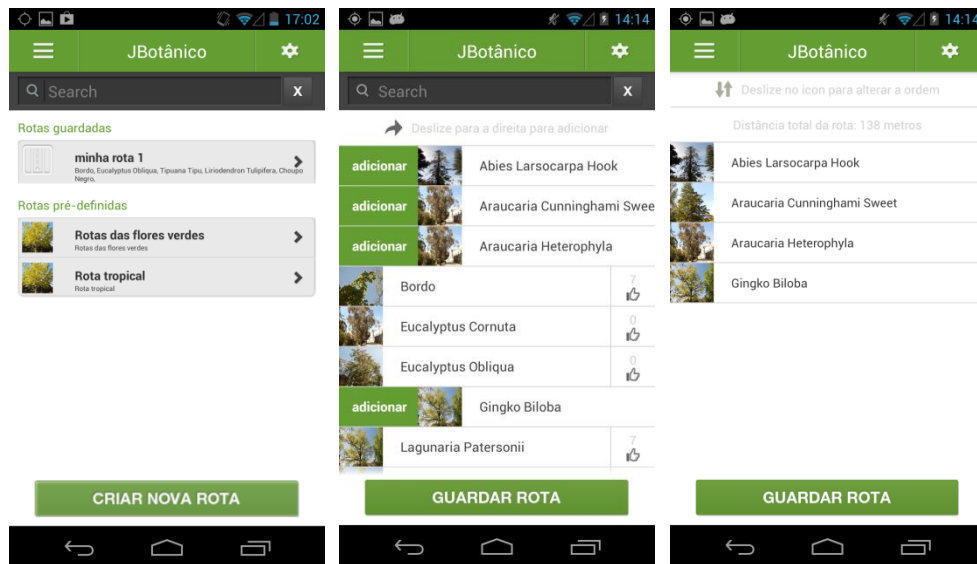


Figura 58: Representação da listagem das rotas (figura da esquerda), da criação de uma nova rota (figura do meio) e da escolha da ordem dos pontos (figura da direita)

A Figura 58 mostra o resultado final para a componente de escolha/criação de rotas. A figura mais à esquerda mostra o ecrã que permite ao utilizador escolher a rota que deseja seguir. É possível ver que o utilizador pode escolher seguir uma rota pré-definida da aplicação ou então uma rota que ele próprio já criou. O utilizador pode também escolher a opção de criar uma nova rota. A figura ao centro mostra o ecrã responsável por

permitir ao utilizador escolher os pontos a adicionar à rota que está a criar. Para a escolha dos pontos, o utilizador tem que deslizar para a direita os itens. É apresentado para cada planta o número de “likes” que esta recolhe. A figura mais à direita mostra o ecrã responsável por permitir ao utilizador alterar a ordem dos pontos a visitar na rota. Para isso, o utilizador deve deslizar os elementos para cima ou para baixo, ordenando assim os pontos. À medida que é alterada a ordem dos pontos, é atualizada a distância total que será percorrida na rota. Este ecrã é apresentado quando o utilizador cria uma nova rota mas também quando este escolhe seguir uma rota que já tinha criado, podendo alterar a ordem dos pontos.

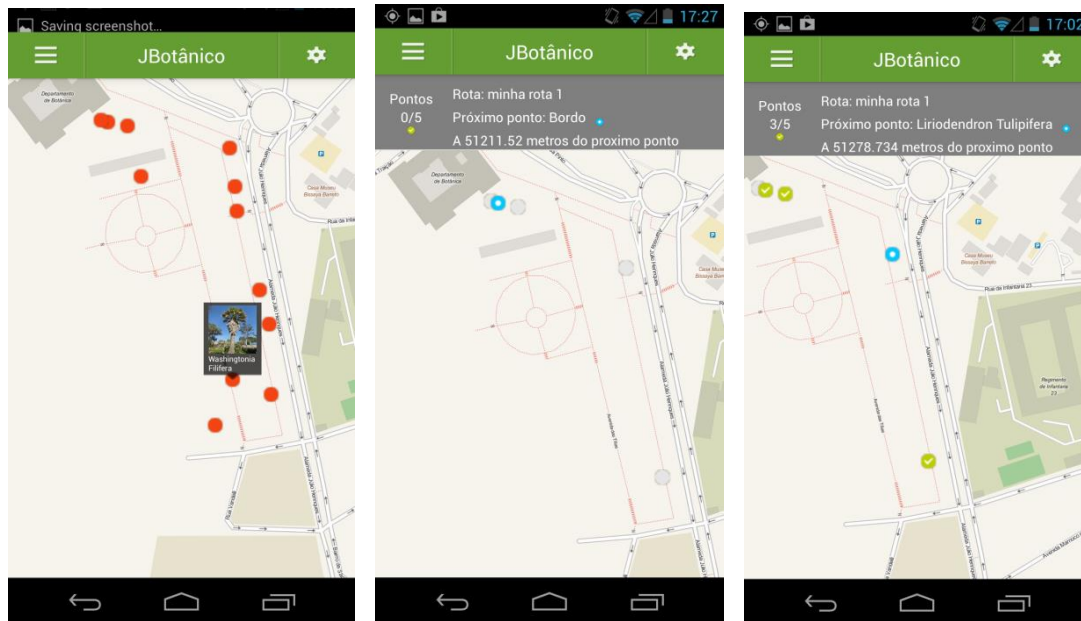


Figura 59: Representação da utilização do mapa 2D na aplicação

A Figura 59 mostra os ecrãs da aplicação aquando a utilização de mapas a duas dimensões. A figura mais à esquerda representa a utilização do mapa no caso geral, apresentando no mapa todas as plantas presentes na aplicação. Os pontos vermelhos representam as plantas. Ao carregar num ponto, é apresentada uma caixa (espécie de *popup*) com um ícone da planta e o seu nome. Carregando no ícone, a aplicação passa para o ecrã de detalhes da planta. As figuras do meio e mais à direita representam a utilização do mapa nas rotas. É apresentado um painel informativo ao utilizador, no topo do ecrã, indicando o número de pontos que já foram vistos, o nome da rota, o nome do próximo ponto a visitar assim como a distância até ao mesmo. De modo a diferenciar os pontos no mapa, o próximo ponto é representado por um ponto azul, os pontos que já foram visitados são representados por pontos verdes e os pontos que faltam visitar são representados por pontos cinzentos. A distância até ao próximo ponto é alterada à medida que o utilizador se desloca, de 10 em 10 metros ou de 10 em 10 segundos.



Figura 60: Resultados para a utilização da realidade aumentada numa rota

A Figura 60 apresenta os resultados obtidos para a utilização da realidade aumentada para seguir uma rota. São apresentadas, no ecrã, as plantas que compõem a rota. Para cada uma delas, é apresentada uma imagem representativa, o nome e a distância a que se encontra do utilizador. A distância é atualizada de 10 em 10 metros ou de 10 em 10 segundos. Os pontos da rota que já foram visitados ficam com o fundo com cor verde. Uma planta considera-se “visitada” sempre que o utilizador visualiza os detalhes da mesma. A próxima planta da rota fica com o fundo de cor azul, como é possível ver na imagem. Na parte superior do ecrã, é apresentada um painel informativo. Este apresenta o número de pontos que já foram vistos, o nome da rota, o nome do próximo ponto da rota assim como a distância que falta até ao mesmo. Observa-se à direita a existência de um “radar” que apresenta as plantas existentes à volta do utilizador sobre a forma de pontos, num alcance de 320 metros à volta da posição do utilizador. Tal como para os ícones, o ponto de cor azul representa o próximo ponto a visitar na rota, os pontos de cor verde representam as plantas que já foram visitadas na rota e os pontos de cor cinzenta representam mas restantes plantas da rota.



Figura 61: Resultados para a utilização da realidade aumentada (guia geral)

A Figura 61 apresenta os resultados obtidos para o guia geral da realidade aumentada. Recorrendo ao guia geral, são apresentadas todas as plantas existentes na aplicação. Para cada uma delas é apresentado um ícone, o nome da mesma assim como a distância até à mesma. Tal como para as rotas, existe um radar no topo superior direito do ecrã que apresenta as plantas existentes num alcance de 320 metros à volta do utilizador, sobre a forma de pontos de cor vermelha.



Figura 62: Resultados para a realidade aumentada na pesquisa de uma determinada planta

A Figura 62 apresenta os resultados obtidos para a procura da localização de uma planta em particular. São apresentadas as mesmas informações que para as rotas e o guia geral.



Figura 63: Resultados para a partilha de dados na rede social Twitter

A Figura 63 apresenta o resultado final para a partilha de dados da rede social Twitter. Na partilha presente na rede social, é apresentada uma frase introduzida pelo utilizador e uma frase gerada pela aplicação móvel que indica a planta que foi visitada. Para além disso, é apresentada uma imagem da planta que foi partilhada.

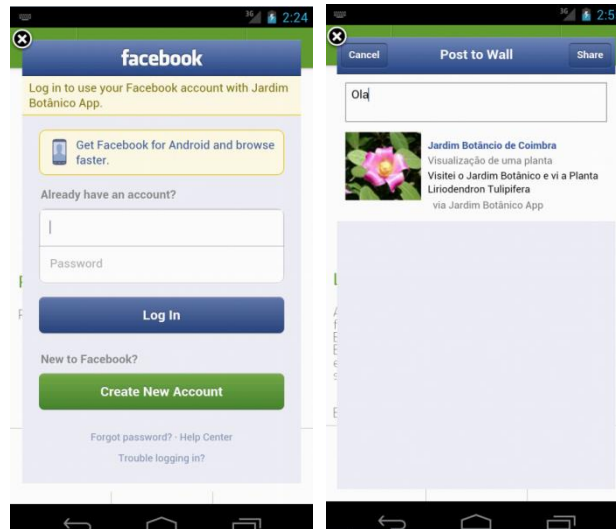


Figura 64: Processo de partilha de informação da rede social Facebook

A Figura 64 apresenta os resultados para o processo de partilha de dados na rede social Facebook. O visitante tem primeiro que fazer *login* na sua conta (figura da esquerda). Caso o *login* seja efetuado com sucesso, é apresentado um formulário onde o utilizador pode escrever uma frase, que será publicada no seu mural do Facebook juntamente com informação adicional sobre a planta partilhada.



Figura 65: Resultado para a partilha de informação na rede social Facebook

A Figura 65 mostra o resultado final, no mural da conta Facebook do visitante, da partilha de dados na rede social Facebook. É apresentada a frase que o visitante inseriu no formulário de partilha juntamente com uma imagem da planta partilhada e o nome da mesma.

Anexo IV - Ecrãs da outra aplicação móvel desenvolvida

Apresentam-se aqui capturas de ecrãs da outra aplicação móvel desenvolvida no decorrer do estágio.

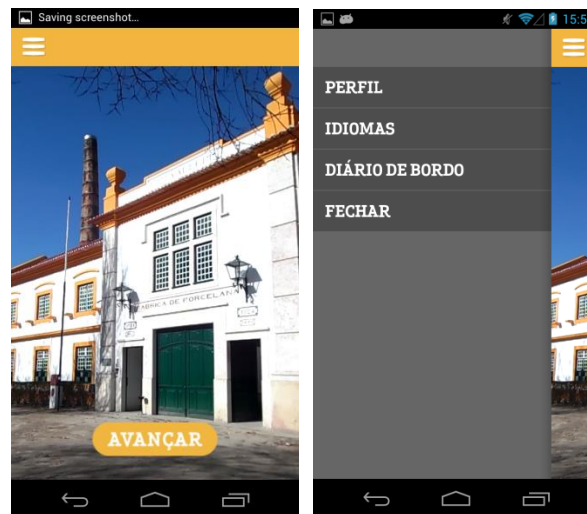


Figura 66: Representação do vídeo introdutório (figura da esquerda) e menu principal (figura da direita)

A Figura 66 representa os ecrãs responsáveis pelo vídeo introdutório da aplicação (figura da esquerda) assim como pelo menu principal da aplicação (figura da direita).

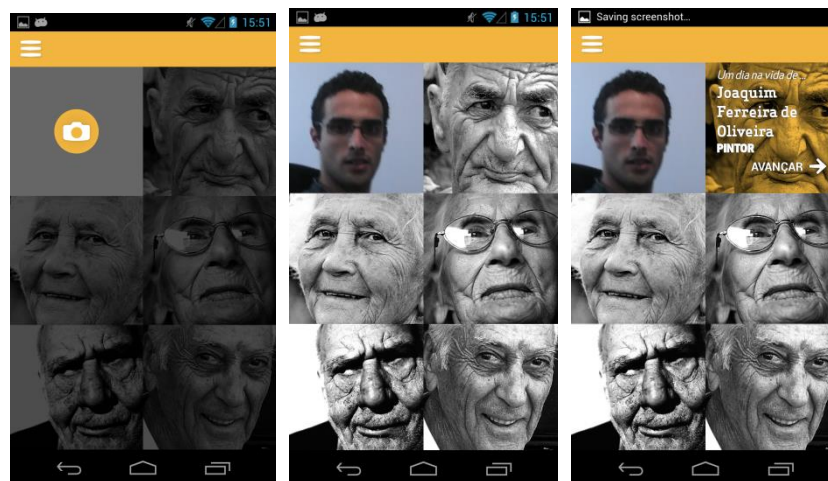


Figura 67: Representação da escolha da personagem a seguir

A Figura 67 representa os ecrãs responsáveis pela escolha da personagem para seguir a sua rota. Inicialmente, as imagens das personagens estão desativadas, para obrigar o utilizador a tirar uma fotografia de modo a criar a sua própria personagem (figura da esquerda). Uma vez tirada a fotografia o utilizador pode escolher uma rota qualquer para seguir (imagens do meio e da direita).

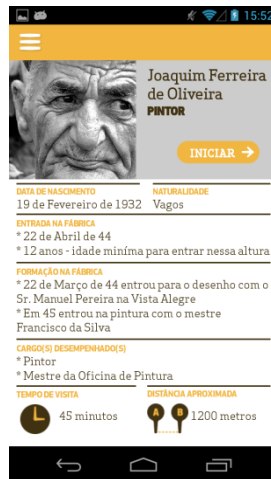


Figura 68: *Printscreen* do BI da personagem escolhida

A Figura 68 representa um *printscreen* do BI da personagem escolhida na grelha de personagem da Figura 67. São apresentadas várias informações sobre a personagem. Carregando no botão inicial é possível iniciar o percurso da personagem escolhida.

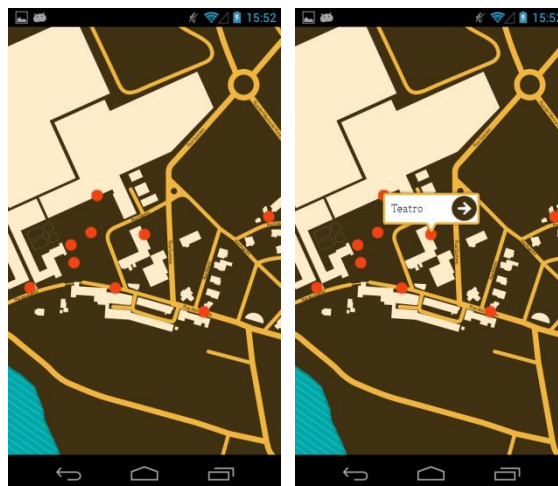


Figura 69: *Printscreen* do mapa utilizado na aplicação

A Figura 69 mostra o mapa da aplicação. Os pontos vermelhos representam os pontos de interesse pertencentes ao percurso escolhido. Carregando num dos pontos, é apresentado ao utilizador uma espécie de “balão” para que, carregando no mesmo, seja apresentada informação multimédia sobre o ponto de interesse.

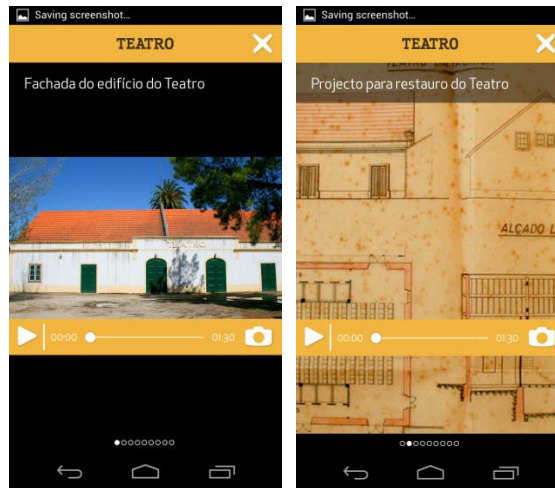


Figura 70: *Printscreen* da galeria de conteúdos multimédia de um ponto de interesse

A Figura 70 representa *printscreens* da apresentação de uma galeria de conteúdos multimédia de um determinado ponto de interesse. Para além de uma galeria de imagens, também é possível reproduzir conteúdos áudio que apresentam informação sobre o ponto de interesse. Também é possível ao utilizador escolher a opção de tirar as suas próprias fotografias, para posteriormente visualizá-las no diário de bordo.



Figura 71: *Printscreen* do diário de bordo da aplicação

A Figura 71 mostra um *printscreen* do diário de bordo da aplicação. O diário apresenta as fotografias tiradas pelo utilizador a um determinado ponto de interesse, agrupando-as por ponto de interesse.

Anexo V – Ecrãs da aplicação de visualização de estatísticas final

Apresentam-se aqui alguns ecrãs da aplicação de visualização de estatísticas depois de desenvolvida.

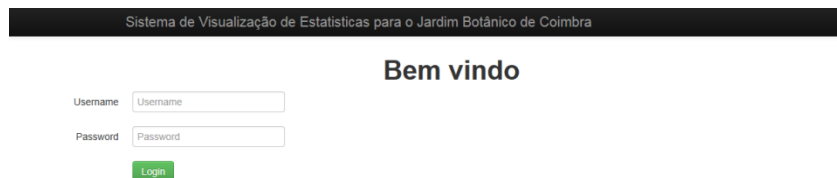


Figura 72: Ecrã de login da aplicação

A Figura 72 representa a página de *login* da aplicação para visualização de estatísticas. Para entrar na aplicação, o utilizador deve introduzir o seu *username* e *password*.

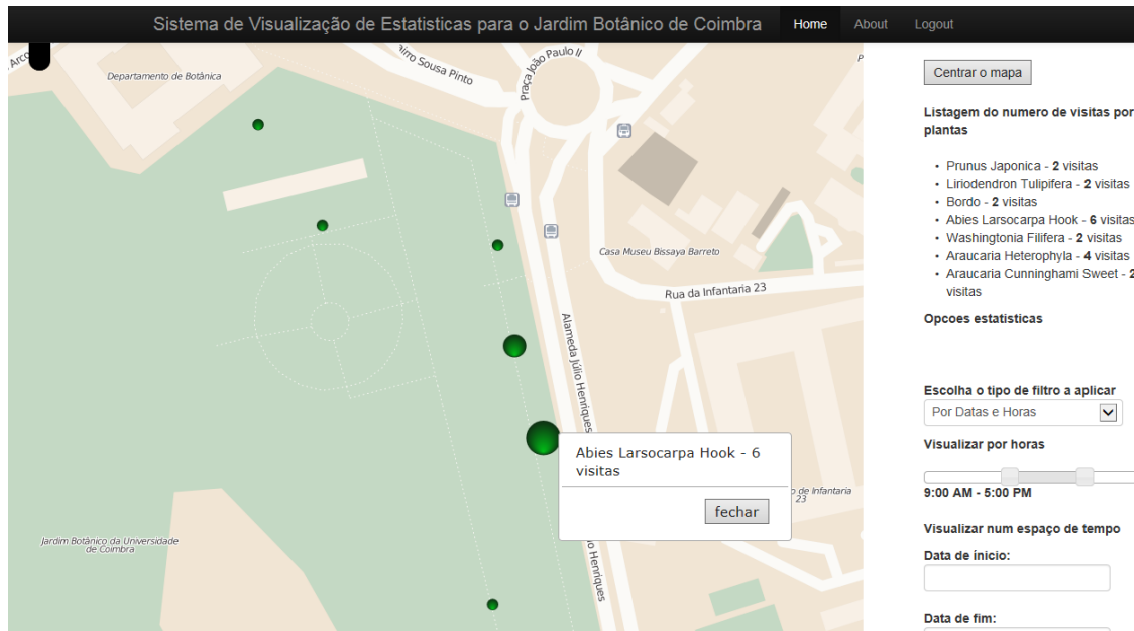


Figura 73: Representação de uma das métricas da aplicação

A Figura 73 representa a apresentação dos resultados para a visualização de uma determinada métrica (o número de visitas por plantas, neste caso). Para as restantes métricas, a interface utilizada é igual à apresentada na Figura 73. Apenas variam as frases que indicam o tipo de métrica apresentada. Observa-se que a página apresenta o mapa do Jardim Botânico assim como esferas de cor verde para representar as plantas, na sua

localização. Consoante o número de visitas às plantas é apresentada uma esfera de maior/menor diâmetro. Ao carregar numa das esferas, é apresentada uma janela informativa com o nome da planta e o número de visitas que esta recolheu. Do lado direito da página, é apresentada uma listagem das plantas apresentando o nome da planta e o número de visitas à mesma. Para além disso, existe um menu que permite efetuar a filtragem dos dados, escolhendo o tipo de filtro a aplicar e os intervalos de tempo. Mais detalhes sobre os filtros podem ser encontrados na Figura 74.

The figure displays three identical panels for filter selection, each titled "Opcoes estatisticas".

- Panel 1:** Dropdown menu: "Por Datas e Horas". Slider: "9:00 AM - 5:00 PM". Date fields: "Data de inicio: 2013-06-24" and "Data de fim: 2013-06-24". Button: "Filtrar".
- Panel 2:** Dropdown menu: "Apenas por Horas". Slider: "9:00 AM - 5:00 PM". Date fields: "Data de inicio:" and "Data de fim:". Button: "Filtrar".
- Panel 3:** Dropdown menu: "Apenas por Datas". Slider: "9:00 AM - 5:00 PM". Date fields: "Data de inicio: 2013-06-01" and "Data de fim: 2013-06-30". Button: "Filtrar".

Figura 74: Representação completa da escolha dos filtros

A Figura 74 mostra como podem ser aplicados filtros aos dados apresentados. O utilizador pode escolher o tipo de filtro a aplicar, podendo escolher filtrar os dados num intervalo de tempo definido por dias e horas simultaneamente, somente por dias ou somente por horas. Na Figura 75 pode ser visualizado o ecrã apresentado após filtragem dos dados.

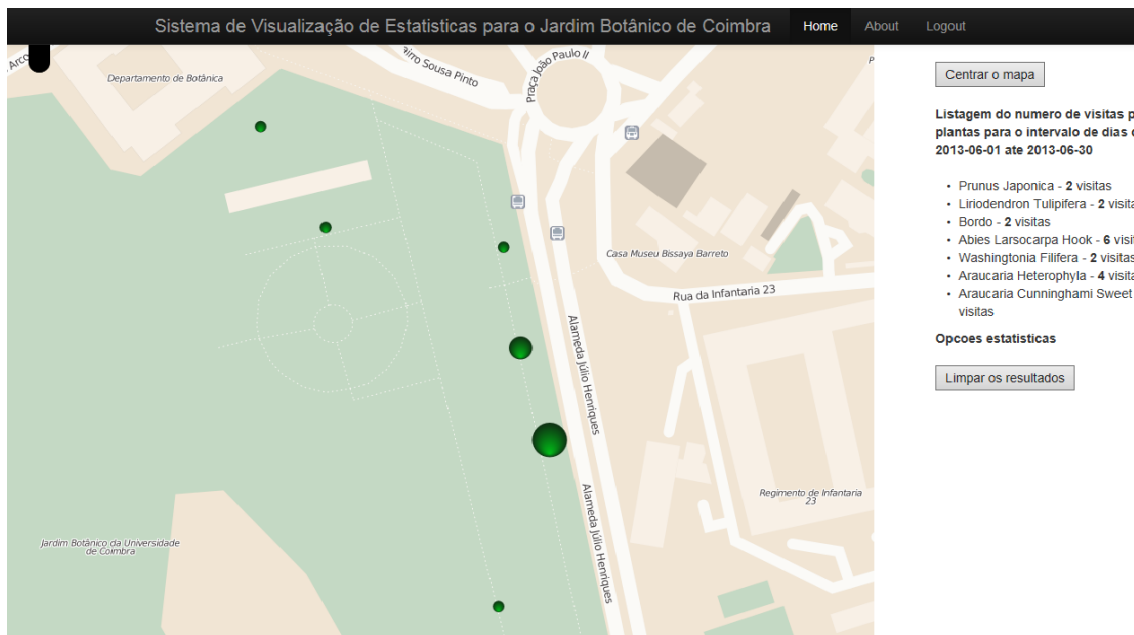


Figura 75: Representação dos resultados para a filtragem dos dados

A Figura 75 mostra a representação da informação após ter aplicado um filtro aos dados. É apresentado ao utilizador uma indicação sobre o tipo de filtro que foi aplicado aos dados assim como a listagem das plantas (nome e número de visitas para cada uma). O utilizador pode escolher limpar os resultados, voltando para os resultados gerais (sem filtro nos dados).

Anexo VI – Chamadas REST dos métodos da API

Segue-se uma listagem do métodos implementados na API existente. Para cada método são apresentados os parâmetros obrigatórios e opcionais (caso existem), assim como o modo pelo qual se efetua a chamada ao método e a resposta devolvida pelo servidor.

Multimedia: São aqui apresentados os métodos para a parte de Multimédia. Apresentam-se os parâmetros de cada método, a chamada ao método e o conteúdo da resposta do método invocado.

Método: Multimédia `{{id}}`

Parâmetros obrigatórios: `{{id}}`. Este parâmetro corresponde ao identificador do recurso multimédia, devolvendo o multimédia correspondente ao `{{id}}` enviado em parâmetro.

Chamada: `http://api.mesh-t.com/api/v1/multimedia/1/?format=json`

Conteúdo da Resposta:

```
{
  author: "",
  count_dislikes: null,
  count_likes: null,
  created_on: "",
  description: "",
  duration: null,
```



```

file_path: "p1_geo_DSC_0003.jpg",
file_size: 140278,
id: "1",
language: "/api/v1/languages/PT/",
location_lat: null,
location_long: null,
name: "Paragem 1 - 1",
position: "",
relative_order: null,
relevance: null,
resolution: "",
resource: "/api/v1/languages/98/",
resource_uri: "/api/v1/multimedia/1/",
thumbnail: "p1_geo_DSC_0003_thumb.jpg",
type: "P",
updated_on: ""
}

```

Método: Multimedia/

Este método devolve todos os conteúdos multimédia presentes na base de dados.

Parâmetros opcionais: {{resource_owner}}. Este parâmetro é opcional e corresponde ao identificador do recurso ao qual corresponde o conteúdo multimédia. Trata-se portanto de uma filtragem de conteúdos multimédia correspondentes a um recurso.

Chamada: <http://api.mesh-t.com/api/v1/multimedia/?format=json>

Resposta:

```

{
  meta: {
    limit: 20,
    next: "/api/v1/multimedia/?offset=20&limit=20&format=json",
    offset: 0,
    previous: null,
    total_count: 76
  },
  objects: [
    {
      author: "",
      count_dislikes: null,
      count_likes: null,
      created_on: "",
      description: "",
      duration: null,
      file_path: "p1_geo_DSC_0003.jpg",
      file_size: 140278,
      id: "1",
      language: "/api/v1/languages/PT/",
      location_lat: null,
      location_long: null,
      name: "Paragem 1 - 1",
      position: "",

```

```

    relative_order: null,
    relevance: null,
    resolution: "",
    resource: "/api/v1/languages/98/",
    resource_uri: "/api/v1/multimedia/1/",
    thumbnail: "p1_geo_DSC_0003_thumb.jpg",
    type: "P",
    updated_on: ""
  },
  {
    ....
  },

```

Resources: apresentam-se aqui os métodos já implementados na API existente para os recursos.

Método: Resources/

Chamada: <http://api.mesh-t.com/api/v1/resources/?format=json>

Resposta:

```

{
  "meta":
  {
    "limit": 20,
    "next": "/api/v1/resources/?offset=20&limit=20&format=json",
    "offset": 0,
    "previous": null,
    "total_count": 31
  },
  "objects":
  [
    {
      "count_dislikes": 0,
      "count_likes": 12,
      "has_indoor_mode": 0,
      "id": 1,
      "parent_event": null,
      "parent_resource": "/api/v1/resources/27/",
      "poi":
      [
      ],
      "relative_order": 0,
      "relevance": 0,
      "rep_image":
      {
        "author": "",
        "count_dislikes": null,
        "count_likes": null,
        "created_on": "",
        "description": "",
        "duration": null,
        "file_path": "http://api.mesh-t.com/media/1.png",
        "file_size": null,
        "id": 1,

```

```

    "language": "/api/v1/languages/PT/",
    "location_lat": null,
    "location_long": null,
    "name": "1.png",
    "parent_multimedia": null,
    "position": "1",
    "relative_order": 0,
    "relevance": 0,
    "resolution": "",
    "resource": "/api/v1/resources/1/",
    "resource_uri": "/api/v1/multimedia/1/",
    "thumbnail": "http://api.mesh-t.com/media/1_thumb.jpg",
    "type": "P",
    "updated_on": ""
  },
  "resource_info":
  [
    {
      "brief_description": "O museu de aveiro, localiza-se no antigo convento de
Jesus, da Ordem Dominicana feminina.",
      "details": "",
      "id": 1,
      "language": "/api/v1/languages/PT/",
      "name": "Entrada do Museu",
      "resource_uri": "/api/v1/resourceinfos/1/"
    }
  ],
  "resource_uri": "/api/v1/resources/1/",
  "service":
  [
  ],
  "type": "ITEM"
},
.....
]]

```

Método: Resources {{id}}

Parâmetros: {{id}}. Este parâmetro corresponde ao identificador do recurso, devolvendo o recurso correspondente ao {{id}} enviado em parâmetro.

Chamada: <http://api.mesh-t.com/api/v1/resources/1/?format=json>

Resposta:

```

{
  "count_dislikes": 0,
  "count_likes": 12,
  "has_indoor_mode": 0,
  "id": 1,
  "parent_event": null,
  "parent_resource": "/api/v1/resources/27/",
  "poi":
  [
  ],

```

```

"relative_order": 0,
"relevance": 0,
"rep_image":
{
  "author": "",
  "count_dislikes": null,
  "count_likes": null,
  "created_on": "",
  "description": "",
  "duration": null,
  "file_path": "http://api.mesh-t.com/media/1.png",
  "file_size": null,
  "id": 1,
  "language": "/api/v1/languages/PT/",
  "location_lat": null,
  "location_long": null,
  "name": "1.png",
  "parent_multimedia": null,
  "position": "1",
  "relative_order": 0,
  "relevance": 0,
  "resolution": "",
  "resource": "/api/v1/resources/1/",
  "resource_uri": "/api/v1/multimedia/1/",
  "thumbnail": "http://api.mesh-t.com/media/1_thumb.jpg",
  "type": "P",
  "updated_on": ""
},
"resource_info":
[
  {
    "brief_description": "O museu de aveiro, localiza-se no antigo convento de
Jesus, da Ordem Dominicana feminina.",
    "details": "",
    "id": 1,
    "language": "/api/v1/languages/PT/",
    "name": "Entrada do Museu",
    "resource_uri": "/api/v1/resourceinfos/1/"
  }
],
"resource_uri": "/api/v1/resources/1/",
"service":
[
],
"type": "ITEM"
}

```

Routes: apresentam-se aqui os métodos já implementados para as rotas.

Método: Routes `{{id}}`

Parâmetros: `{{id}}` . Este parâmetro corresponde ao identificador da rota, devolvendo a rota correspondente ao `{{id}}` enviado em parâmetro.

Chamada: <http://api.mesh-t.com/api/v1/routes/1/?format=json>

Resposta:

```
{
  description: "",
  elements: [
    "/api/v1/route_elements/40/",
    "/api/v1/route_elements/41/",
    "/api/v1/route_elements/42/",
    "/api/v1/route_elements/43/",
    "/api/v1/route_elements/44/",
    "/api/v1/route_elements/45/",
    "/api/v1/route_elements/46/"
  ],
  id: "1",
  language: "/api/v1/languages/PT/",
  name: "Percurso Pedestre",
  requested_time: 1334744280.209146,
  resource_uri: "/api/v1/routes/1/"
}
```

Método: Routes/

Chamada: <http://api.mesh-t.com/api/v1/routes/?format=json>

Resposta:

```
{
  "meta":
  {
    "limit": 20,
    "next": "/api/v1/routes/?offset=20&limit=20&format=json",
    "offset": 0,
    "previous": null,
    "total_count": 32
  },
  "objects":
  [
    {
      "count_dislikes": 0,
      "count_likes": 0,
      "elements":
      [
        "/api/v1/route_elements/4/",
        "/api/v1/route_elements/5/",
        "/api/v1/route_elements/6/",
        "/api/v1/route_elements/7/",
        "/api/v1/route_elements/8/",
        "/api/v1/route_elements/9/",
        "/api/v1/route_elements/10/",
        "/api/v1/route_elements/11/",
        "/api/v1/route_elements/12/"
      ],
      "id": 1,
      "parent_resource": "/api/v1/resources/27/",
      "relative_order": 0,
    }
  ]
}
```

```

    "relevance": 0,
    "resource_uri": "/api/v1/routes/1/",
    "route_info":
    [
      {
        "brief_description": "",
        "description": "O percurso monumental pelo antigo Convento de Jesus mostra
os espaços conventuais que sobreviveram até aos nossos dias. São, na sua maior parte,
espaços sagrados, uma vez que as áreas privadas desapareceram devido às sucessivas
obras de adaptação deste edifício a colégio e, posteriormente, a museu. Desapareceram as
celas, a enfermaria, a botica, as oficinas e os celeiros. Sobreviveram ainda alguns vertigios
das cozinhas na área da atual cafeteria.
",
        "id": 1,
        "language": "/api/v1/languages/PT/",
        "name": "Percurso Monumental",
        "resource_uri": "/api/v1/routeinfos/1/"
      }
    ]
  },

```

Método: route_elements/
Chamada: http://api.mesh-t.com/api/v1/route_elements/?format=json
Parâmetros opcionais: {{parente_route}}. Este parâmetro é opcional e corresponde ao identificador da rota à qual pertence o elemento. Trata-se portanto de uma filtragem de elementos pertencentes a uma determinada rota.

Resposta:

```

{
  "meta":
  {
    "limit": 20,
    "next": "/api/v1/route_elements/?offset=20&limit=20&format=json",
    "offset": 0,
    "previous": null,
    "total_count": 315
  },
  "objects":
  [
    {
      "element_order": 1,
      "id": 4,
      "resource":
      {
        "count_dislikes": 0,
        "count_likes": 12,
        "has_indoor_mode": 0,
        "id": 1,
        "parent_event": null,
        "parent_resource": "/api/v1/resources/27/",
        "poi":
        [

```

```

    "relative_order": 0,
    "relevance": 0,
    "rep_image":
    {
      "author": "",
      "count_dislikes": null,
      "count_likes": null,
      "created_on": "",
      "description": "",
      "duration": null,
      "file_path": "http://api.mesh-t.com/media/1.png",
      "file_size": null,
      "id": 1,
      "language": "/api/v1/languages/PT/",
      "location_lat": null,
      "location_long": null,
      "name": "1.png",
      "parent_multimedia": null,
      "position": "1",
      "relative_order": 0,
      "relevance": 0,
      "resolution": "",
      "resource": "/api/v1/resources/1/",
      "resource_uri": "/api/v1/multimedia/1/",
      "thumbnail": "http://api.mesh-t.com/media/1_thumb.jpg",
      "type": "P",
      "updated_on": ""
    },
    "resource_info":
    [
      {
        "brief_description": "O museu de aveiro, localiza-se no antigo convento de
Jesus, da Ordem Dominicana feminina.",
        "details": "",
        "id": 1,
        "language": "/api/v1/languages/PT/",
        "name": "Entrada do Museu",
        "resource_uri": "/api/v1/resourceinfos/1/"
      }
    ],
    "resource_uri": "/api/v1/resources/1/",
    "service":
    [
    ],
    "type": "ITEM"
  },
  "resource_uri": "/api/v1/route_elements/4/",
  "route": "/api/v1/routes/1/",
  "type": "ITEM"
},

```