



Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia da
Universidade de Coimbra

Posicionamento em plataformas móveis

Tese de Mestrado em Engenharia Informática
Relatório Final

Daniel Filipe Isidoro Azeiteiro
dfia@student.dei.uc.pt

Orientador: Doutor Pedro Furtado

30 de Agosto de 2012

Resumo

Dado o forte crescimento dos dispositivos móveis, destacando-se os populares *smartphones* e *tablets*, a sua adopção tem-se tornado cada vez maior por parte dos utilizadores, tornando-se objectos que automatizam e facilitam algumas das tarefas que o ser humano realiza no seu dia a dia. A troca de qualquer tipo de informação é bastante simples com uso destes dispositivos, sendo uma destas trocas de informação a partilha da posição actual do utilizador, algo que se tornou bastante popular hoje em dia, não apenas pelos desenvolvimentos dos mecanismos de localização e dispositivos indicadores de caminhos e rotas, mas também pelas redes sociais. Este trabalho consiste na construção de um sistema de partilha de posições entre dispositivos móveis, através da criação de grupos e actividades que relacionam os utilizadores entre si. O seu objectivo é permitir uma partilha automática e personalizada de posições durante um determinado período de tempo, definido como actividade. Esta partilha poderá ser efectuada utilizando diferentes protocolos de comunicação e deverá ser suportada por diferentes sistemas operativos. Os resultados finais deste projecto apresentam-se sobre a forma de uma aplicação pronta a ser instalada num dispositivo móvel, bem como na construção de uma plataforma *web* que sirva toda a aplicação e os clientes que utilizem a aplicação no seu dispositivo móvel.

Palavras-chave: partilha de posições, computação ubíqua, mobilidade, geolocalização, automatização, sistemas baseados em localização

Conteúdo

1	Introdução	10
1.1	Projecto e Equipa	10
1.2	Contexto e Motivação	11
1.3	Objectivos	12
1.4	Estrutura do Documento	13
2	Estado da Arte	15
2.1	Soluções	15
2.1.1	<i>Foursquare</i>	15
2.1.2	<i>Life360</i>	16
2.1.3	<i>Glympse</i>	16
2.1.4	<i>Google Latitude</i>	17
2.1.5	<i>Gowalla</i>	17
2.1.6	Comparação	18
2.2	Tecnologias	20
2.2.1	Sistema Operativo Móvel	20
2.2.2	Serviços de mapas	22
2.2.3	Geolocalização	23
3	Especificação e Análise de Requisitos	27
3.1	Justificação do Sistema a Implementar	27
3.2	Cenários	28
3.3	Arquitectura do Sistema	29
3.3.1	Descrição da Arquitectura	29
3.3.2	Identificação de Requisitos	33
3.3.3	Especificação das Tecnologias Utilizadas	37
4	Abordagem	42
4.1	Metodologia de Desenvolvimento	42
4.2	Descrição Geral da Arquitectura	43
4.3	Base de dados	44
4.4	Módulos Implementados	47
4.4.1	Visualização de Mapas	47

4.4.2	Mapas Personalizados	49
4.4.3	Criação e visualização de Rotas	50
4.4.4	<i>Login</i> e Registo	51
4.4.5	Operações de Grupos e Actividades	51
4.4.6	Visualização de actividades	52
4.4.7	Sincronização	53
4.4.8	API	53
4.4.9	Servidor	54
4.5	Carga de Testes	54
4.5.1	Cenário de Testes	55
4.5.2	Testes Não-Funcionais	56
4.5.3	Testes Funcionais	57
5	Reflexões e trabalho futuro	61
5.1	Conclusão da dissertação	61
5.2	Trabalho Futuro	62
5.3	Reflexão sobre o desenrolar dos trabalhos	62

Lista de Tabelas

2.1	Tabela de comparação entre soluções existentes	18
4.1	Tabela de comparação entre plataformas de mapas	48
4.2	Especificações do Servidor	55
4.3	Especificações do dispositivo <i>Android</i>	55
4.4	Requisitos não funcionais segundo a implementação	56
4.5	Requisitos funcionais segundo a implementação	57
4.5	Requisitos funcionais segundo a implementação	58
4.5	Requisitos funcionais segundo a implementação	59
4.5	Requisitos funcionais segundo a implementação	60

Lista de Figuras

3.1	Cenário de utilização do projecto	29
3.2	Arquitectura do Servidor	30
3.3	Arquitectura do Servidor Temporário	32
3.4	Arquitectura do Cliente	32
4.1	Fluxo Servidor - Cliente	44
4.2	Diagrama da Base de Dados	45

Tabela de Acrónimos

Access Point	Dispositivo que permite a ligação de outros dispositivos <i>wi-fi</i> a uma rede
A-GPS	<i>Assisted GPS</i>
API	<i>Application Programming Interface</i>
ASCII	<i>American Standard Code for Information Interchange</i>
Check-in	Acto de notificar a chegada a um determinado ponto de interesse
CISUC	Centro de Investigação e Sistemas da Universidade de Coimbra
DEI	Departamento de Engenharia Informática
Delay	Atraso na comunicação entre dois pontos
Download	Transferência de ficheiros de uma localização remota
GPS	<i>Global Positioning System</i>
GPU	<i>Graphics Processing Unit</i>
GSM	<i>Global System for Mobile Communications</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JPEG	<i>Joint Photographic Experts Group</i>
JSON	<i>JavaScript Object Notation</i>
LAMP	<i>Linux-Apache-MySQL-PHP</i>
LBS	<i>Location-based Services</i>
Market Share	Quota de mercado
Offline	Não possui ligação para o exterior
OPS	<i>Open Street Maps</i>
PNG	<i>Portable Network Graphics</i>
PHP	<i>Hypertext Preprocessor</i>
POI	Ponto de Interesse

REST	<i>Representational state transfer</i>
RFID	Radio-frequency identification
SDK	<i>Software Development Kit</i>
SGDB	Sistema de Gestão de Base de Dados
SMS	<i>Short Message Service</i>
SQL	<i>Structured Query Language</i>
SSE	<i>Software and Systems Engineering</i>
SSL	<i>Secure Sockets Layer</i>
UC	Universidade de Coimbra
UI	<i>User Interface</i>
Upload	Envio de ficheiros para uma localização remota
URL	<i>Uniform Resource Locator</i>
Web Mapping	Implementação e geração de mapas na <i>World Wide Web</i>
Wi-Fi	Mecanismo que permite a criação de uma rede sem fios
WMS	<i>Web Map Service</i>
WWW	<i>World Wide Web</i>

Capítulo 1

Introdução

Este relatório foi elaborado no âmbito do estágio de Mestrado de Engenharia Informática do Departamento de Engenharia Informática (DEI) da Universidade de Coimbra tendo sido dividido em dois semestres. Este relatório contempla todo o trabalho desenvolvido durante esses semestres, onde foram definidos o planeamento e os requisitos necessários à implementação do projecto, procedendo-se à posterior definição de arquitectura do sistema, implementação da arquitectura e testes finais de sistema.

1.1 Projecto e Equipa

O projecto a desenvolver está integrado no grupo de investigação Software and Systems Engineering (SSE) [35], o qual faz parte do Centro de Informática e Sistemas da Universidade de Coimbra (CISUC).

O CISUC é um centro de investigação em informática e telecomunicações, ficando situado no Departamento de Engenharia Informática da Universidade de Coimbra.

Por sua vez o SSE é um dos grupos do CISUC com áreas de investigação ligadas a Sistemas Confiáveis, Gestão de dados e Sistemas distribuídos e paralelos. O SSE encontra-se sediado no DEI da Universidade de Coimbra, dividido em três laboratórios, cada um deles ligado a uma das áreas de investigação citadas anteriormente, fazendo parte da equipa um conjunto de Professores do departamento e também alunos de doutoramento.

A equipa de desenvolvimento deste trabalho inclui o autor deste relatório, Daniel Azeiteiro como orientando, o Doutor Pedro Furtado como orientador e o Mestre José Cecílio como colaborador, sendo os dois últimos membros do grupo de investigação SSE, de onde surgiu a iniciativa de desenvolvimento do projecto.

Este projecto enquadra-se num projecto de investigação do Engenharia Informática do Departamento de Engenharia Informática (DEI) da Universidade de Coimbra (UC). A logística e as infra-estruturas de suporte ao projecto serão

garantidas pelo DEI da UC.

1.2 Contexto e Motivação

Com o desenvolvimento cada vez mais rápido dos dispositivos móveis e a sua utilização cada vez mais comum na sociedade actual surge a necessidade de utilização de novas ferramentas nestes dispositivos que possibilitem a introdução de novas funcionalidades a actividades já existentes. Actividades práticas do dia a dia, como por exemplo uma ida às compras, pode hoje ser feita com uma lista de compras colocada num *smathphone*, as leituras de preços simplesmente efectuadas com a leitura do código de barras do produto através da câmara do dispositivo, podendo mesmo esta funcionalidade, em alguns casos, criar um carrinho de compras virtual que no final é apresentado para posterior pagamento e entrega dos produtos escolhidos. A partilha de posições foi outra funcionalidade bastante desenvolvida nos últimos tempos, sendo hoje possível partilhar quase de modo instantâneo as coordenadas do local onde o utilizador se encontra ou mesmo o cálculo de um caminho até um determinado ponto ou local de interesse, algo que há poucos anos atrás só seria possível com a utilização de um mapa em papel e com a busca manual dos caminhos nesse mesmo mapa.

Actualmente são muitas as aplicações que tornam o dia a dia dos utilizadores mais simples, automatizando muitas das tarefas diárias, tornando-as por vezes mais interessantes e divertidas. Os *smartphones* têm vindo a criar novas dinâmicas de interacção entre indivíduos, contribuindo para o fortalecimento de relações através de redes sociais e outras aplicações semelhantes.

No entanto, apesar das novas tecnologias existem actividades que o ser humano não dispensa, não podendo estas ser realizadas por máquinas. Um exemplo destas actividades são as actividades desportivas, que muitas das vezes envolvem inúmeros participantes nas mais diversas áreas.

Deste modo questionou-se como seria possível integrar o mundo dos *smartphones* com a prática de actividades desportivas ou outro tipo de actividades envolvendo diversos participantes.

As actividades desportivas estas são muitas das vezes realizadas em grupo, pelo que seria interessante qualquer participante saber qual a posição de um colega num determinado momento. Assim, foi idealizada uma aplicação que permitisse a partilha de posições entre utilizadores que realizassem determinadas actividades em conjunto, durante determinados períodos de tempo.

Esta partilha poderia ser efectuada de diversos modos diferentes, conforme a actividade a ser realizada. A existência de grupos, como por exemplo grupos de amigos, é uma ideia que também se torna interessante, uma vez que pessoas num grupo de amigos tendem a realizar diversas actividades em conjunto.

Um exemplo prático deste tipo de actividade pode ser uma actividade des-

portiva como um passeio de bicicleta ou a pé, em que a qualquer momento um participante que deseje entrar na actividade ou se afaste do grupo possa saber a localização dos seus colegas utilizando a aplicação. Outro exemplo que não implica a prática desportiva são por exemplo as noites académicas da cidade de Coimbra: muitas das vezes os estudantes saem à noite em grupo, no entanto para poderem encontrar os restantes colegas são obrigados a telefonar ou enviar uma SMS e esperar pela resposta. Um uso prático seria cada um dos estudantes colocar a aplicação em funcionamento de modo a realizar, por exemplo, de 20 em 20 minutos, uma actualização da sua posição actual. Os restantes colegas apenas necessitam de aceder à aplicação e verificar o local onde os colegas se encontram.

1.3 Objectivos

O principal objectivo deste estágio centra-se no desenvolvimento de uma aplicação que permita a partilha automática de posições entre os utilizadores, de uma forma automática, durante determinados períodos de tempo. Definiu-se assim a criação de um projecto que visa criar um sistema personalizado de partilha de posições, suportado pelos dispositivos móveis actuais e que fosse algo interessante e inovador de modo a cativar a atenção dos utilizadores.

Aquando da definição do projecto, pensou-se também em não restringir a aplicação a algo específico mas sim em construir algo genérico, isto é, que não integrasse apenas actividades desportivas mas que pudesse ser utilizado como uma aplicação genérica adaptável a diversos cenários e situações. Para dar um exemplo podemos por exemplo indicar uma empresa de camionagem, em que os camiões representariam os participantes e os fretes que cada um desempenhava as diferentes actividades existentes, sabendo-se assim sempre a partir da torre de controlo a posição de cada um dos camiões da empresa.

Deste modo pretende-se que a aplicação a desenvolver seja dotada de funcionalidades como:

- Partilha automática de posições entre grupos de utilizadores;
- Visualização num mapa das posições dos utilizadores;
- Personalização dos mapas, com elementos como rotas ou mapas personalizados;
- Diferentes protocolos de comunicação;
- Funcionamento sem necessidade de ligação de dados;
- Suporte para os sistemas operativos móveis actuais;
- Fácil utilização;

A partilha automática de posições entre grupo de utilizadores consiste na partilha das coordenadas cartesianas de cada utilizador para com os restantes. Pretende-se que esta operação seja efectuada dentro de um grupo restrito de utilizadores e não com toda a aplicação, de forma a garantir a privacidade dos seus utilizadores. De modo ao próprio utilizador poder visualizar estas posições é também necessário que a aplicação apresente um mapa onde serão apresentadas as posições dos restantes utilizadores do grupo de forma a garantir uma melhor interpretação da posição de cada um. Por forma a enriquecer esta partilha este projecto deve também permitir a personalização destas partilhas com a adição de mapas e rotas personalizados pelos utilizadores, permitindo assim algo mais inovador e interessante do que apenas a utilização de um mapa normal.

A funcionalidade de diferentes protocolos de comunicação passa por não restringir a comunicação da aplicação a uma simples ligação de dados *web*, mas sim permitir a sua expansão a outros protocolos já existentes como o GSM, garantindo assim uma maior liberdade de utilização. O funcionamento sem necessidade de ligação de dados permite que os utilizadores possam usufruir da aplicação quando não possuem qualquer ligação de dados, não ficando assim esta sem funcionamento quando o utilizador perde o acesso de dados.

O suporte para os sistemas operativos móveis mais utilizados vem proporcionar uma maior oportunidade para a aplicação se expandir no mercado pois ao ser desenvolvida para os sistemas com mais utilizadores e o facto de não ficar restrita a apenas um sistema operativo móvel é um passo importante o seu sucesso futuro. Do mesmo modo o facto de ser de fácil utilização é outro dos factores que podem levar ao sucesso da mesma pois aplicações complexas têm normalmente menos tendência a ser adoptadas pelos utilizadores.

Estas funcionalidades serão estudadas e implementadas, sendo todo este processo explicado neste relatório.

1.4 Estrutura do Documento

De modo a facilitar a leitura e compreensão deste relatório optou-se pela divisão do mesmo em cinco capítulos distintos.

O presente capítulo tem como objectivo familiarizar o leitor com o contexto do trabalho efectuado e da problemática que nele é abordada, sendo também definidos os objectivos que se pretendem atingir com o mesmo.

No Estado da Arte (capítulo 2) é realizado um estudo detalhado sobre os conceitos teóricos envolvidos neste trabalho. Serão abordadas as tecnologias existentes que serão utilizadas para resolver os problemas do projecto realizando também um estudo de soluções semelhantes já existentes no mercado.

Com a abordagem (capítulo 4), são apresentadas as decisões tomadas na implementação do sistema. É aqui descrita a análise de requisitos efectuada para o desenvolvimento do projecto, a especificação das tecnologias utilizadas e a arquitectura do sistema.

A implementação da abordagem proposta é descrita no capítulo 4 onde são apresentados os passos efectuados na implementação do projecto, segundo a arquitectura definida para o mesmo e os testes efectuados sobre aquilo que foi desenvolvido. No final deste capítulo são apresentadas os diferentes testes realizados e os respectivos resultados, com o intuito de validar a abordagem proposta.

Por fim, o capítulo 5 apresenta as conclusões obtidas ao longo do desenvolvimento deste trabalho, sendo apresentadas algumas ideias que futuramente poderão ajudar numa melhoria e do sistema proposto e da abordagem implementada.

Capítulo 2

Estado da Arte

Serve o presente capítulo para descrever o estado da arte relativamente às tecnologias e soluções em que este projecto se enquadra. Classificando o projecto a desenvolver como uma aplicação direccionada para dispositivos móveis, tendo como principal objectivo a interacção entre utilizadores e a partilha das suas respectivas posições, foi realizado um estudo relativamente às soluções, projectos e tecnologias presentes no mercado.

Será primeiramente apresentado um estudo com as soluções mais semelhantes actualmente disponíveis no mercado, de forma a garantir que a aplicação a desenvolver será original e única. Posteriormente serão apresentadas as tecnologias disponíveis que são necessárias para a criação da aplicação, efectuando uma comparação das mesmas, de modo a seleccionar aquelas que melhor se adequam a este trabalho.

2.1 Soluções

Antes de iniciar o estudo das plataformas móveis e os requisitos da aplicação, foram estudadas as soluções actualmente disponíveis no mercado de modo a conhecer a oferta disponível. São apresentadas as aplicações consideradas mais relevantes bem como uma breve descrição dos objectivos de cada uma e funcionalidades oferecidas.

2.1.1 *Foursquare*

O *Foursquare* [12] é actualmente uma das ferramenta mais populares de partilha de localização entre pessoas. Possui na sua base de dados um conjunto de pontos de interesse, pontos estes que cada utilizador é livre de criar, definindo o seu tipo e características (como morada, categorias, comentários e sugestões). O

objectivo é que cada utilizador indique a posição onde se encontra, fazendo *check-in* no ponto de interesse que se encontra mais próximo. Estes pontos de interesse são introduzidos maioritariamente pelos utilizadores pelo que podem variar entre moradias privadas ou centros comerciais, criados estes últimos pelas próprias empresas de modo a publicitar os seus estabelecimentos. Em termos de partilha, um utilizador possui um grupo de amigos na sua lista pessoal com os quais partilha a sua localização cada vez que realiza um *check-in*, sendo esta partilha extensível a redes sociais como o *Twitter* [37] ou *Facebook* [10], se o utilizador assim o desejar.

Existe ainda um *ranking* entre os utilizadores, podendo estes obter pontos conforme a quantidade de *check-ins* realizados, o número de visitas a cada POI, entre outros. Para além das funcionalidades já mencionadas, contém ainda uma lista de pontos de interesse que são sugeridos ao utilizador consoante a sua popularidade e proximidade. Esta aplicação encontra-se disponível para os sistemas operativos móveis mais populares e é também possível a sua utilização através de um *web browser* que suporte sistema de localização, o que permite uma maior flexibilidade de utilização, tornando a aplicação utilizável em qualquer plataforma.

2.1.2 *Life360*

A aplicação *Life360* [26] foi desenvolvida com o objectivo de manter os membros de uma família em constante contacto de modo a que cada um deles consiga determinar a posição de outro quando assim o desejar. Este software permite o registo de um grupo, podendo ou não ser uma família, que irá realizar a partilha de posições entre si. A partilha é realizada de modo semelhante ao *Foursquare*, isto é, através da realização de *check-ins* nos locais onde o utilizador se encontra. Um dos pontos-fortes desta aplicação é a possibilidade de um utilizador conseguir realizar um pedido a outro para que este faça *check-in* no local onde se encontra. Isto torna mais simples o modo de localizar outras pessoas do grupo uma vez que o utilizador, que deseja saber a informação acerca de outro, tem a possibilidade de o notificar e não se limitar a esperar. Esta aplicação está disponível para os principais sistemas operativos móveis actuais e também através de *web browser* que permite uma maior flexibilidade de uso ao utilizador.

2.1.3 *Glympse*

O *Glympse* [13] é um software que permite ao utilizador partilhar a sua localização com outros utilizadores, durante o período de tempo que desejar. A partilha da localização realiza-se com as pessoas que o utilizador desejar, isto é, com a quantidade de contactos que pretender adicionar à actividade que vai realizar. Todos os contactos adicionados receberão um alerta, quer por *e-mail* ou SMS com a informação da partilha da localização e o respectivo URL. Os utilizadores

convidados a visualizar a partilha não necessitam de qualquer registo no sistema ou software específico, bastando apenas um *web browser* de modo a conseguir aceder ao URL, sendo mostrado posteriormente um mapa onde é possível visualizar todo o trajecto realizado pelo utilizador que enviou o convite. A aplicação permite também a integração com o *Facebook* possibilitando a importação de contactos e partilha de informação no mural. Necessita de uma aplicação específica para ser realizada a partilha de posição, existindo já para os dispositivos móveis mais utilizados com excepção de *Symbian*.

2.1.4 *Google Latitude*

O *Google Latitude* [15] é o produto apresentado pela *Google* concorrente aos serviços de partilha de localização. O seu objectivo é bastante semelhante ao *Foursquare* no entanto possui menos funcionalidades. Este software permite realizar *check-ins* por parte do utilizador nos locais onde este se encontra, no entanto, se o ponto de interesse não existir na base de dados da *Google* não é possível ao utilizador adicionar esse ponto. A actualização da posição também é realizada automaticamente ao contrário da maioria dos *softwares*, possibilitando o *check-in* de modo automático, se o optar utilizador por activar esta funcionalidade. Caso o utilizador não defina a sua privacidade, a aplicação actualiza a posição do utilizador entre determinados períodos de tempo de modo automático. Um dos pontos fortes desta aplicação é a integração com os diversos serviços fornecidos pela *Google* de modo automático e sem a necessidade de quaisquer configurações. São disponibilizadas aplicações nativas para os principais sistemas operativos móveis actuais e também é dada a possibilidade de utilização desta ferramenta num *web browser*.

2.1.5 *Gowalla*

Esta aplicação [20] é bastante cujo funcionamento se assemelha bastante ao *Foursquare*. Os seus métodos de funcionamento são bastante semelhantes: cada utilizador possui um grupo de amigos com os quais partilha a sua localização. Possui aplicação nativa para a maior parte dos dispositivos móveis actuais e uma versão web com menos funcionalidades. Apresenta alguns guias turísticos das cidades mais populares de modo a realizar visitas guiadas ao utilizador, se este assim o desejar.

Foi adquirido pelo *Facebook* em Dezembro de 2011 [11] com o objectivo de integrar este serviço com o serviço de locais do *Facebook*. Até ao momento ainda não foi lançada qualquer nova versão desta aplicação por parte do *Facebook Places* nem mesmo a integração dos seus pontos de interesse nos serviços fornecidos pelo mesmo.

2.1.6 Comparação

Na Tabela 2.1 as diversas funcionalidades que cada uma das aplicações descritas anteriormente de forma a serem mais explícitas as diferenças entre as mesmas. De referir que devido a não ter sido fornecido um nome específico ao projecto a desenvolver é apresentado na tabela com o nome de *Projecto* como forma de distinção dos restantes.

	<i>Foursquare</i>	<i>Life360</i>	<i>Glympse</i>	<i>Latitude</i>	<i>Gowalla</i>	<i>Projecto</i>
Multi-Plataforma	✓	✓	✓	✓	✓	✓
<i>Compatibilidade com Browser</i>	✓		✓	✓	✓	✓
<i>Ligação Web</i>	✓	✓	✓	✓	✓	✓
<i>Ligação GSM</i>						✓
<i>Check-ins</i>	✓	✓		✓	✓	
<i>Auto-Update</i>			✓			✓
<i>POI's</i>	✓			✓	✓	
<i>Amigos</i>	✓		✓	✓	✓	
<i>Grupos</i>			✓			✓
<i>Actividades</i>		✓	✓		✓	✓
<i>Mapas personalizados</i>						✓
<i>Rotas personalizadas</i>						✓
<i>Modo offline</i>						✓

Tabela 2.1: Tabela de comparação entre soluções existentes

Ao analisarmos a Tabela 2.1 verificamos que não existe nenhuma aplicação que apresente os requisitos propostos por este projecto, apresentando, no entanto, algumas delas funções semelhantes.

Analisando os campos da tabela verificamos que todas as aplicações estudadas são desenvolvidas para múltiplos sistemas operativos pois actualmente o mercado móvel encontra-se equitativamente distribuído por diferentes sistemas operativos. Esta é uma característica que a aplicação desenvolvida também abrange, sendo criada com o objectivo de ter a máxima integração possível com todos os dispositivos de mercado, não restringindo aqui apenas o sistema operativo onde será executada mas também as características dos mesmos, como por exemplo o poder de processamento e memória do dispositivo do utilizador.

O facto do *software* possuir a funcionalidade de compatível com um *browser* é uma mais-valia para este projecto ser utilizado num dispositivo para o qual não foi desenvolvida qualquer aplicação nativa ou mesmo permitir o acesso às

funcionalidades do sistema sem a necessidade de instalação de *software*. Com o aparecimento do HTML5 [21] é possível através de um *browser* usufruir de diversos serviços como a geolocalização, a utilização *offline* de páginas *web* e até mesmo de pequenas bases de dados, tudo isto sem a necessidade da instalação de *software* adicional.

Os tipos de conexão suportados pelas aplicações estudadas são apenas conexões de dados ao servidor. A aplicação desenvolvida contempla também um envio via GSM que possibilitará o funcionamento, mesmo na ausência de ligação de dados, permitindo que o utilizador utilize este protocolo para a troca de posições via SMS, não necessitando assim de uma ligação de dados que por vezes está associada a custos elevados de utilização.

A funcionalidade de *check-in* não se encontra presente na aplicação desenvolvida pois segundo o conceito do projecto o utilizador não necessita de indicar a sua posição, sendo esta actualizada de forma automática no decorrer de uma actividade. Assim a funcionalidade *auto-update* surge em substituição da anterior, possibilitando ao utilizador uma utilização da aplicação sem a necessidade de estar constantemente a interagir com o dispositivo para a actualização da sua posição.

A utilização de pontos de interesse não foi contemplada na aplicação desenvolvida. Não sendo do interesse a visita de pontos de interesse mas sim a partilha de posições a inclusão deste tipo de dados não é relevante numa primeira fase do projecto.

Do mesmo modo a funcionalidade *Amigos* também não é abrangida, apesar de o ser em algumas das aplicações concorrentes. A utilização de grupos e actividades em que os utilizadores se possam inscrever substitui a necessidade da criação deste tipo de serviço. Deste modo foi implementada a criação de grupos, onde vários utilizadores se poderiam inscrever e também actividades, que fazem parte de um determinado grupo. Este tipo de restrição permite assim que os utilizadores se possam ligar por grupos, criando as suas actividades específicas que são independentes e não visíveis por outros utilizadores fora desse grupo.

As opções *Mapas Personalizados* e *Rotas Personalizadas* são também exclusivas do projecto, uma vez que não foi encontrada nenhuma aplicação onde fosse possível carregar um mapa personalizado sobre o mapa principal, assim como rotas personalizadas, que apesar de serem passíveis de serem criadas em algumas das aplicações, não é possível durante a execução da aplicação mostrar as mesmas sobre o mapa principal.

Por fim o modo de funcionamento *offline* é uma das características mais ausente nas aplicações estudadas, pois sem uma ligação de dados a sua utilização é bastante limitada.

2.2 Tecnologias

Neste capítulo serão apresentadas as tecnologias servem como base para o desenvolvimento da aplicação pretendida. Para este tipo de aplicação é importante dissertar sobre tecnologias necessárias a uma plataforma colaborativa de posicionamento, sendo fundamentais as análises aos sistemas operativos móveis existentes, serviços de mapas e sistemas de geolocalização.

2.2.1 Sistema Operativo Móvel

Para a análise dos sistemas operativos móveis deste trabalho foram escolhidos aqueles que actualmente mais se destacam no mercado e cuja utilização tende a aumentar. Para isso foram estudados aqueles que apresentam um *market share* mais elevado. Ao analisarmos o gráfico de distribuição de quota de mercado dos últimos dois anos [27] verificamos que existem três que se destacam perante os restantes: *Symbian* [36], *Android* [2], *iOS* [23].

O *Symbian* é o sistema operativo pertencente à *Nokia* [29], que adquiriu este sistema operativo em Dezembro de 2008. Antes de ser adquirido na sua totalidade este sistema era partilhado por diversas marcas de dispositivos móveis, sendo uma evolução de um sistema operativo que surgiu no final dos anos oitenta, denominado por EPOC. Este sistema foi evoluindo ao longo dos anos, tendo sido renomeado para *Symbian* em 2001, quando foi lançado para o mercado em Junho, com o *Nokia 9210 Communicator*, atingindo um elevado sucesso devido à introdução de uma nova UI, o que levou a um elevado crescimento até aos dias de hoje, sendo ainda um dos maiores líderes de mercado.

Com a aquisição do *Symbian* por parte da *Nokia*, este tornou-se um sistema maioritariamente utilizado por esta mesma marca para os seus dispositivos, o que levou a *Nokia* a disponibilizar um SDK que permite o desenvolvimento de aplicações para esta plataforma, suportando uma diversidade de linguagens. Este sistema operativo foi desenhado com base em três filosofias: manter a integridade e segurança dos dados dos utilizadores, não desperdiçar tempo ao utilizador e que todos os recursos existentes são escassos. Estas filosofias permitiram o acompanhamento da evolução dos dispositivos móveis, mantendo este sistema na liderança. Recentemente com o aparecimento de novos sistemas operativos como o *Android* e *iOS* o *Symbian* foi bastante abalado tendo perdido bastante da sua quota de mercado. A recente aliança entre a *Nokia* e a *Microsoft* [30] levou ao abandono do *Symbian* na América do Norte, tendo substituído o seu sistema operativo pelo novo sistema operativo *Windows Phone 7* [39], tentando assim unir os produtos de ambas as empresas com o objectivo de conquistar quota de mercado face os novos sistemas operativos móveis *Android* e *iOS* que estão a crescer rapidamente e a diminuir o número de utilizadores da plataforma *Symbian*.

Anteriormente à junção com o *Windows Phone 7* a *Nokia* lançou a sua própria loja de aplicações denominada *Ovi Store* que permitia o acesso a aplicações *Symbian* aos clientes deste sistema. Contudo, esta loja virtual nunca conseguiu atingir

um elevado número de aplicações, perdendo a atenção dos programadores que focaram o desenvolvimento das suas aplicações em sistemas como o *Android*, *iOS* e *Windows Phone*.

O sistema operativo *Android* [2] é um sistema operativo baseado em UNIX pertencente à *Google*, tendo sido adquirido pela mesma em 2005 e lançado em 20 de Setembro de 2008. Sendo um sistema operativo com um Kernel baseado em UNIX é um sistema desenvolvido com princípios de estabilidade, segurança e robustez, permitindo que as suas aplicações sejam executadas num ambiente à parte, que não tem acesso aos recursos do sistema, a não ser com a permissão do próprio utilizador do dispositivo.

Em termos de desenvolvimento a *Google* disponibiliza um SDK bastante completo, com um emulador do sistema operativo compatível com *Windows*, *Linux* e *Mac OS X*. O desenvolvimento destas aplicações é maioritariamente realizado na linguagem *Java*, sendo interpretado no sistema operativo móvel por uma máquina virtual designada de *Dalvik* [3] que permite a interpretação do código Java nas suas aplicações.

Outro dos pontos fortes deste sistema é a integração com os serviços *Google* [14] que permite ao utilizador uma sincronização dos dados dos dispositivos móvel com os serviços *Google* bem como a disponibilização da maioria dos serviços da empresa em aplicações para o sistema operativo. A criação do *Android Market*, mais recentemente renomeada de *Play Store*, como uma loja virtual de aplicações, onde é permitido aos programadores a submissão das suas aplicações depois de aprovação, torna este sistema escolhido por muitos devido a uma menor complexidade do desenvolvimento e facilidade de publicação das aplicações.

Actualmente, ao contrário do *Symbian*, este sistema operativo é distribuído por diversas empresas nos seus dispositivos móveis, o que tem levado a um crescimento de mercado bastante elevado por parte do *Android*.

O *iOS* é o sistema operativo móvel da *Apple*, tendo sido apresentado ao público a 9 de Janeiro de 2007, juntamente com o *iPhone*. Inicialmente apresentado como um sistema operativo *Mac OS* para *iPhone*, foi renomeado posteriormente para *iOS* com a criação do *iPod Touch* e *iPad*. São actualmente os únicos dispositivos onde é permitida a execução do *iOS*. Do mesmo modo, a *Apple* disponibiliza um SDK bastante completo para o *iOS*, no entanto só é possível executar o mesmo e desenvolver para este sistema utilizando o sistema operativo *Mac OS X*.

Este sistema caracteriza-se pela sua simplicidade e facilidade de utilização, bem como estabilidade e robustez, pois assim como o *Android* o seu *Kernel* é baseado no sistema operativo UNIX. O facto de o seu *hardware* ser limitado aos dispositivos fabricados pela *Apple* pode ser considerado um dos pontos fortes do sistema, uma vez que o *software* é produzido tendo sempre em conta um *hardware* específico. Assim como os dispositivos anteriores, também o *iOS* é acompanhado com uma loja de aplicações, sendo esta actualmente uma das que possui uma

maior quantidade de aplicações [34]. O sucesso da *App Store*, nome dado pela *Apple* à sua loja de aplicações, deve-se em parte ao sucesso já conseguido pela *iTunes Store*, a qual a também foi igualmente incluída no sistema operativo móvel.

Uma das maiores limitações deste sistema operativo foi a incapacidade de lidar com *multitasking*. A *Apple* resolveu este problema apresentado na versão 4 do *iOS* sendo possível a partir desta versão de sistema a execução de diversas aplicações em simultâneo.

Actualmente, ao compararmos os três sistemas operativos móveis mencionados, verificamos que o *Android* e o *iOS* possuem um crescimento bastante mais acentuado que o *Symbian*. Ao analisarmos os mais recentes dados de vendas [4] verificamos que existe uma grande adesão aos dois sistemas, sendo o seu futuro bastante promissor, o que não acontece com o *Symbian* que, apesar da aliança entre a *Nokia* e a *Microsoft* e a sua ainda actual liderança de mercado, não demonstra um crescimento tão acentuado como os dois anteriores.

2.2.2 Serviços de mapas

Sendo este projecto uma ferramenta que integra serviços de localização, os quais são fortemente integrados com a visualização em mapas, foram analisados os serviços de mapas disponíveis para serem utilizados na aplicação pretendida.

Estes serviços, denominados como *Web Mapping*, permitem a visualização de mapas cartográficos na *World Wide Web*. Através da junção de diversas fotografias armazenadas numa base de dados, capturadas por satélites espalhados ao longo do planeta, torna-se possível visualizar num mapa virtual de qualquer parte do mundo, podendo aumentar a escala do mapa de modo a ver em mais profundidade a zona pretendida, caso essa zona se encontre abrangida pelos satélites de captura, isto é, caso existam fotografias capturadas mais detalhadamente pelos satélites nesse ponto.

Com o avanço desta tecnologia foram adicionadas novas funcionalidades a este serviço como a geração de rotas que indicam ao utilizador os caminhos disponíveis para se deslocar de um ponto até outro, a inclusão de POI's nos mapas que permitem ao utilizador encontrar mais facilmente o que deseja quando se encontra num local desconhecido e, mais importante, a adaptação desta tecnologia aos dispositivos móveis. Com a inclusão deste tipo de serviços nos dispositivos móveis passou também a ser possível a utilização destes mapas em modo *offline*, necessitando o utilizador apenas de descarregar toda a informação sobre o mapa desejado para o seu dispositivo para posterior utilização quando assim o desejar. Interessa assim para a aplicação pretendida dissertar sobre tecnologias que estejam já bem adaptadas aos dispositivos móveis uma vez que são o principal alvo de desenvolvimento.

Das tecnologias existentes, duas delas destacaram-se para serem utilizadas no

projecto: o *Open Street Maps* [32] e o *Google Maps*.

O *Open Street Maps* surgiu no ano 2004 com o objectivo de criar uma plataforma *open-source* de mapas através da qual qualquer pessoa era livre quer para actualizar os dados desta plataforma com novas rotas, quer para utilizar os dados existentes na base de dados do *Open Street Maps*. Inicialmente um grupo de voluntários disponibilizou as primeiras rotas apresentadas, continuando a crescer com a adesão de empresas como a *Automotive Navigation Data* [8] que doou mapas de estradas da Holanda bem como da China e Índia.

Actualmente esta ferramenta é um dos *softwares open-source* mais utilizados, contado já com aplicações nas plataformas móveis mais utilizadas como o *Android* e *iOS*. O facto de se tratar de um projecto colaborativo e *open-source* foram algumas das características que permitiram o desenvolvimento deste sistema.

O *Google Maps* é o sistema de mapas da *Google* que surgiu em Março de 2004 com o nome de *Google Local* [16]. Rapidamente foi desenvolvido e em 2005 a *Google* implementou novas funcionalidades como a vista de satélite e direcções, bem como a extensão do serviço em novos países. Neste mesmo ano foi criada a API o que possibilitou que novas aplicações surgissem utilizando esta ferramenta. A evolução fez com que esta se estendesse aos dispositivos móveis permitindo aos utilizadores obter informação da sua localização mesmo quando o sinal GPS era inexistente.

Esta constante inovação de serviços fornecidos pela plataforma da *Google* colocou-a na liderança do mercado de *Web Mapping*. Actualmente existem diversas funcionalidades do serviço como por exemplo a vista de rua, vista do céu, vista de oceanos e mesmo a vista de outros planetas como é o caso de Marte. Estas novas funcionalidades, actualmente inexistentes, elevaram ainda mais a popularidade do serviço e a disponibilização de uma API de acesso aos dados fez com que seja o serviço de escolha de muitos programadores.

Ao analisarmos o serviço do ponto de vista de aplicações móveis, existem aplicações nativas lançadas pela *Google* para este mercado. São disponibilizadas aplicações para os principais sistemas operativos móveis actuais, bem como uma versão *web* optimizada para dispositivos móveis que tornam o serviço funcional, incluindo a própria geolocalização, sem a necessidade da instalação de nenhum *software* adicional.

2.2.3 Geolocalização

Os sistemas de geolocalização permitem a descoberta da localização específica de um determinado objecto ou mesmo de uma rota específica. Trata-se de uma serviço essencial ao funcionamento do projecto de modo a que seja possível obter

a localização do utilizador num determinado momento. Este termo abrange diversas áreas, mas no entanto, para o contexto do nosso problema vamos apenas focar-nos em *Location-based Services* (LBS) que permitem obter uma localização aproximada em termos geográficos, isto é, coordenadas geográficas de um dado objecto.

O sistema mais preciso e actualmente em uso designa-se por GPS, tendo sido criado em 1973 com o objectivo de ultrapassar as limitações dos sistemas de posicionamento até então existentes. Foi desenvolvido pelo departamento de defesa do Estados Unidos da América tendo-se tornado completamente operacional apenas em 1994.

Este sistema funciona com base em satélites que orbitam o planeta Terra destinados apenas para este fim. Cada um destes satélites possui uma localização específica e um relógio que são as ferramentas base para o funcionamento do serviço. Um cliente que pretenda utilizar este serviço deverá possuir um receptor GPS que tem como objectivo a ligação a estes satélites de modo a obter a localização actual. Este receptor apenas tem como funcionalidade receber os sinais uma vez que todo o processo de determinação de posição é realizado pelos satélites. Os satélites comunicam entre si através da utilização de um relógio interno, calculam a distância a que se encontram entre si, conseguindo criar um mapa em três dimensões quando são conhecidas as posições de quatro ou mais satélites. Assim o cliente recebe estas informações dos satélites e, através de cálculos matemáticos da distância a que se encontra dos mesmos, consegue obter a sua localização com uma precisão de cerca de vinte metros.

Existem algumas limitações a este serviço: devido ao cálculo de posições ser realizado através do sinal proveniente dos satélites em órbita e o número destes não ser muito elevado (actualmente existem vinte e quatro em funcionamento) torna-se difícil a um receptor adquirir sinal de pelo menos quatro destes satélites de modo a determinar a sua posição. Descartando falhas como erros no relógios internos ou a existência de um algum *delay* na resposta, podemos afirmar que o maior problema enfrentado pela tecnologia GPS passa pela necessidade de recepção dos sinais de pelo menos quatro satélites para obter uma posição correcta, agravando-se este problema quando o receptor se encontra com obstáculos no seu caminho como por exemplo a existência de prédios ou objectos numa cidade que impedem a recepção do sinal.

Numa tentativa de ultrapassar algumas das limitações do sistema GPS foi criado o *Assisted GPS* (A-GPS) que tenta melhorar o tempo de recepção dos dados dos satélites GPS, utilizando para isso uma ligação de dados como por exemplo GPRS ou 3G, auxiliando o dispositivo no cálculo das coordenadas de posicionamento (daí o nome de GPS assistido). Um sistema A-GPS pode funcionar em dois tipos diferentes de categorias:

- A informação é usada para obter satélites mais rapidamente, sendo fornecido ao receptor GPS a informação de órbita dos satélites GPS para que

estes possam ser mais facilmente localizados e também fornecendo tempos mais precisos;

- O cálculo da posição é feito pelo servidor, usando informação enviada pelo receptor GPS: o receptor GPS apenas capta um pequeno sinal GPS, com um tempo aproximado da posição, enviando esta para o servidor para posteriormente ser processada pelo mesmo. Por seu lado o servidor de assistência tem um bom sinal GPS e pode comparar os sinais recebidos com os seus, conseguindo uma localização precisa. O conhecimento de coordenadas anteriores e dos pontos de posicionamento das torres emisoras (*cell towers*) permite também um cálculo mais preciso da posição e das condições iossféricas onde se encontra o dispositivo.

Para além destas vantagens este sistema permite ainda reduzir o processamento durante o cálculo, utilizando uma tecnologia denominada *MS-Assisted* que permite que os cálculos necessários para a determinação da posição sejam realizados do lado do servidor de assistência. Para além disto muitos sistemas A-GPS combinam ainda tecnologias de posicionamento *wi-fi* e triangulação de células, que serão explicados posteriormente neste relatório, para obter uma localização mais rápida.

O sistema GPS normal consegue obter a primeira localização em aproximadamente 30 a 40 segundos, no entanto necessita de obter a informação de órbita dos satélites para conseguir calcular a posição corrente do dispositivo. Sendo a sua taxa de transferência de cerca de 50 b/s, leva algum tempo a obter toda a informação necessária, para além de que se existir uma falha na comunicação, toda a informação é perdida e o processo é recomeçado do ponto de partida. Por seu lado no A-GPS, existe um servidor da parte do operador de rede que possui toda a informação necessária acerca do posicionamento dos satélites numa base de dados local. Assim o A-GPS ao realizar a ligação a estes servidores consegue taxas de transferência bastante mais elevadas, utilizando uma ligação de dados *wi-fi* ou 3G.

Como referido anteriormente, existem ainda outros tipos de serviços de localização bastante utilizados como é o caso do posicionamento calculado através de redes *wi-fi*. Esta tecnologia funciona de um modo semelhante ao GPS com a diferença que ao invés do uso de satélites são utilizados como pontos de referência os emissores *wi-fi* existentes no raio de alcance do receptor. Algumas empresas mantêm grandes bases de dados com as localizações de *access points* de modo a que quando um cliente se encontra num raio de alcance de uma dessas redes seja possível um cálculo estimado da sua posição, dependendo da distância a que este se encontra do emissor.

Esta tecnologia permite ultrapassar alguns dos problemas do GPS, nomeadamente o problema da ausência de sinal no interior de edifícios e outros locais isolados das emissões exteriores, uma vez que as redes *wi-fi* existem cada vez em

maior número por todo o mundo. Possui algumas limitações como a necessidade de uma ligação de dados ao servidor de modo a obter a informação da posição acerca do *access point* que se encontra ligado e dos seus vizinhos. Outra das limitações deste sistema é a inexistência de informação do *access point* na base de dados do servidor principal, pois caso o servidor de localização não tenha qualquer informação acerca deste ou dos seus vizinhos é impossível determinar a posição do utilizador nestas condições.

Semelhante ao método *wi-fi* temos ainda um método que determina uma posição aproximada do utilizador através da antena GSM a que este se encontra conectado. Este sistema baseia-se na força do sinal com que o dispositivo se encontra conectado às antenas GSM, permitindo assim estabelecer um raio de localização do dispositivo à volta da antena, uma vez que a localização da própria antena já é conhecida pelo operador. No geral não é um método muito preciso devido ao longo alcance deste tipo de antenas, no entanto, em zonas urbanas onde o número de antenas deste tipo é bastante mais elevado, é possível realizar uma triangulação do sinal do dispositivo entre várias antenas deste tipo e chegar a uma localização bastante mais aproximada da posição do utilizador.

Muitos dos métodos utilizados hoje em dia para determinação de posições são métodos híbridos, utilizando uma junção das tecnologias descritas acima de modo a obter uma posição o mais definida possível, tal como foi referido anteriormente com o modelo A-GPS.

Capítulo 3

Especificação e Análise de Requisitos

Neste capítulo será apresentada uma especificação da arquitectura do sistema que constitui a aplicação, os cenários de validação do sistema, bem como a infra-estrutura de suporte. A especificação basear-se-á na lista de requisitos apresentados anteriormente. Serão apresentadas as tecnologias estudadas bem como a justificação da escolha das mesmas. As tecnologias serão divididas em tecnologias presentes do lado do Servidor 3.3.1 e posteriormente serão apresentadas as tecnologias presentes na aplicação Cliente 3.3.1.

Serão também apresentados dois cenários possíveis de utilização do projecto e justificada a escolha dos sistemas operativos móveis para os quais serão realizadas aplicações nativas.

3.1 Justificação do Sistema a Implementar

Na presente secção será fundamentada a escolha do tipo de sistema a implementar e a razão da implementação de uma nova aplicação perante as já existentes.

Aplicações Móveis Nativas Para o desenvolvimento do projecto houve a necessidade de escolha de quais os sistemas operativos móveis para que se iriam desenvolver aplicações nativas uma vez que se torna impraticável o desenvolvimento para todas as plataformas móveis existentes. Tendo como base o tempo disponível para a execução do projecto decidiu-se desenvolver o projecto numa primeira fase para a plataforma Android, juntamente com uma plataforma web que permite a utilização da aplicação através de um *browser*.

A escolha destes sistemas foi baseada nos argumentos apresentados no capítulo do Estado da Arte (ver cap. 2), nos quais foram estudados os sistemas operativos móveis mais utilizados actualmente, sendo destacados apenas três. Destes

optou-se por não desenvolver para plataforma *Symbian* devido à sua baixa actividade no mercado actual relativamente aos dois sistemas escolhidos. O facto de actualmente ainda se encontrar na liderança de mercado torna-o uma plataforma com grande potencial, no entanto, segundo as análises realizadas, verificamos que continua cada vez mais a perder para *Android* e *iOS* e, nem mesmo a aliança com a *Microsoft* que se mostrava bastante promissora, não conseguiu até ao momento obter resultados significativos. Deste modo, e segundo os estudos realizados, escolheram-se para o desenvolvimento da aplicação os sistemas operativos móveis *Android* e *iOS* para os quais se pretende desenvolver aplicações nativas para as suas plataformas.

Destes dois sistemas optou-se por iniciar o desenvolvimento da plataforma *Android*, juntamente com a plataforma *web* e a criação dos respectivos serviços. O sistema operativo *iOS* foi deixado para desenvolvimento posterior a estes e, devido à escassez de tempo, não foi possível implementar uma aplicação nativa para esta plataforma, sendo contudo o projecto desenvolvido com vista a ser posteriormente implementada uma aplicação nativa para esta plataforma.

Serviço de Mapas e Localização Ambos os sistemas apresentados, *Google Maps* e *Open Street Maps* mostram-se bastante promissores, contudo o uso da plataforma de mapas da *Google* como sistema de mapas apresentado por omissão nos sistemas operativos móveis escolhidos para o desenvolvimento torna esta a seleccionada. O conjunto de serviços que complementam esta plataforma e a sua documentação são também pontos fortes ao uso desta tecnologia. Serão ainda apresentados alguns testes realizados com a plataforma *Open Street Maps* no capítulo Abordagem (ver cap. 4) de forma a garantir a escolha mais adequada. Relativamente ao serviço de localização optou-se pela escolha de um sistema de implementação híbrido, que combina as tecnologias A-GPS e *wi-fi* de forma a garantir uma localização mais precisa, mesmo quando uma das tecnologias não se encontra disponível.

3.2 Cenários

O projecto a desenvolver poderá ser implementado em diversos tipos de cenários e arquitecturas, serão assim apresentados dois tipos de cenários distintos para os quais a aplicação será integrada.

Nos cenários apresentados será contemplado o uso da aplicação, na qual existirá um servidor principal onde os clientes se conectam para realizar actividades.

A Figura 3.1 representa um esquema do projecto em execução com a existência de um servidor principal. Neste cenário qualquer utilizador que se encontre numa actividade a decorrer procederá à troca de informações com o servidor principal.

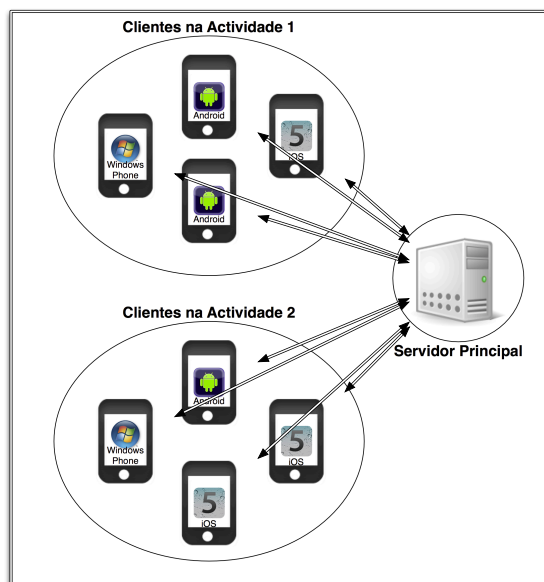


Figura 3.1: Cenário de utilização do projecto

Todo o tráfego é registado pelo servidor principal e os registos das actividades serão armazenados na base de dados principal do sistema. Cada cliente estará identificado com a actividade para a qual está registado, trocando periodicamente informações sobre as posições dos restantes clientes dessa mesma actividade. Nenhum cliente comunica directamente com os restantes, sendo toda a comunicação realizada por intermédio do servidor da aplicação. Esta comunicação poderá ser realizada quer através de uma conexão *web* ou uma conexão GSM, que serão explicadas posteriormente.

3.3 Arquitectura do Sistema

Na presente secção é descrita a arquitectura do sistema implementado, a especificação das tecnologias utilizadas e a descrição da infra-estrutura de suporte.

3.3.1 Descrição da Arquitectura

A natureza deste projecto implica o desenvolvimento de um modelo cliente-servidor no qual os clientes serão todos os dispositivos móveis que fazem uso da aplicação e o servidor todos os módulos responsáveis por servir os pedidos desses mesmos clientes. Deste modo será desenvolvido um conjunto de módulos que permitam a utilização de serviços por partes dos clientes.

Servidor O esquema do servidor pode ser visualizado na Figura 3.2.

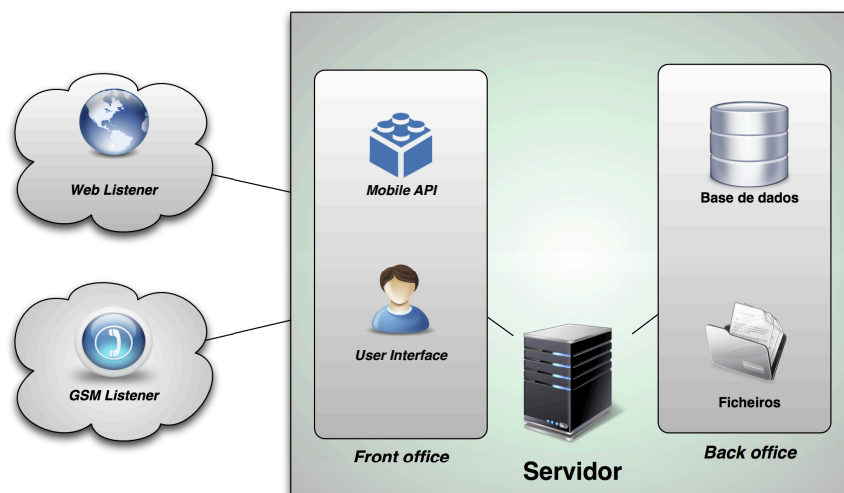


Figura 3.2: Arquitectura do Servidor

O servidor da aplicação apresentará no seu *front office* dois tipos de interface: uma interface que permita a utilização *web* da aplicação e uma API, responsável pelas respostas aos pedidos realizados pelos clientes *mobile*. A administração do sistema deverá ser possível com uma conta de administrador de sistema que irá permitir a um utilizador, com os devidos privilégios, ter acesso ao servidor do sistema e realizar a administração de todos os dados da aplicação, como os seus utilizadores, permissões e manuseamento dos dados guardados.

A interface de utilizador estará disponível ao utilizador comum, sendo desenvolvida numa plataforma *web* para a qual os detalhes técnicos serão especificados posteriormente neste relatório. O utilizador deverá efectuar um registo ou então um *login*, conforme se encontre ou não já registado, sendo posteriormente redireccionado para o ecrã principal da aplicação, onde terá acesso a todas as funcionalidades da aplicação. Esta interface deverá permitir ao utilizador a criação de actividades e grupos, bem como a edição e eliminação dos mesmos. Também será disponibilizada uma ferramenta que permita o desenho de rotas personalizadas pelo utilizador, assim como o envio de mapas personalizados que se irão sobrepor aos mapas originais do serviço de mapas. Será ainda possibilitado aos clientes a consulta de um mapa com as respectivas posições dos utilizadores das actividades correntes, permitindo assim também a participação em actividades por parte dos clientes que utilizam este tipo de interface.

Esta plataforma servirá igualmente os clientes móveis para os quais não se desenvolveu uma aplicação nativa, como o *Symbian* por exemplo, possibilitando assim a utilização da aplicação em múltiplas plataformas.

Quanto ao *back office* este será constituído por uma base de dados que guar-

dará todos os dados da aplicação. Nesta base de dados deverão constar todos os dados referentes aos utilizadores, actividades, grupos, posições e rotas, bem como as referências aos ficheiros existentes. O sistema de ficheiros será utilizado para armazenar todo o conteúdo referente a mapas personalizados, uma vez que cada mapa personalizado será guardado sobre a forma de ficheiro de imagem. Estes ficheiros são carregados apenas uma vez, se necessário, para cada dispositivo que utilize a aplicação pelo que as escritas e leituras de ficheiro em disco não serão muito intensas num cenário de normal utilização.

Relativamente aos *listeners* que são apresentados na Figura 3.2 estes representam os dois diferentes tipos de conexão que é possível realizar para com o servidor. O módulo *web* representa a comunicação por omissão, para a qual estão disponíveis todas as interfaces referidas anteriormente, requerendo esta uma conexão *web* da parte dos clientes. O GSM *listener* representa outro tipo de comunicação que será possível realizar ao servidor, realizando comunicações utilizando o protocolo GSM via SMS. A implementação deste protocolo permite aos clientes que utilizem a aplicação através de um dispositivo móvel com capacidade GSM que troquem informações com o servidor principal utilizando este protocolo, mais concretamente através da troca de SMS's entre cliente e servidor com as suas posições. Será também utilizado o armazenamento interno do dispositivo para guardar dados referentes a actividades, numa base de dados como o caso de mapas personalizados e ficheiros de configuração e a base de dados interna do dispositivo para armazenar dados de utilizador, actividades e grupos. Este segundo protocolo vem assim permitir a utilização parcialmente *offline* da aplicação, não sendo necessário o uso de uma conexão *web* para a sua utilização.

Do ponto de vista da recolha de dados apenas se optou pela recolha da latitude e longitude de cada posição do utilizador. Discartaram-se dados como a altitude do utilizador visto que a aplicação foi pensada para uma utilização onde nem sempre existe um valor preciso da posição do utilizador (quando se adquirem dados geográficos a partir de redes *wi-fi* ou quando o sinal GPS se encontra bloqueado por edifícios dentro de uma cidade, por exemplo) não fazendo assim sentido, numa primeira versão da aplicação a recolha de todos estes dados.

Servidor Temporário Em determinadas actividades deverá ser possível tornar um dos dispositivos móveis que não participa na actividade num servidor da mesma. A Figura 3.3 representa a arquitectura de um servidor temporário.

Para este servidor funcionar correctamente será necessário uma sincronização de dados com o servidor principal. Esta sincronização de dados deverá ser efectuada antes da actividade e não durante a actividade, não só devido à limitação de recursos destes dispositivos mas também à incoerência de dados que pode existir se tudo não estiver correctamente sincronizado no início de uma actividade. A sincronização deve também ser referente apenas à actividade para a qual se deseja criar o servidor, sendo necessário apenas a sincronização dos dados dos clientes da aplicação e dos dados da actividade em causa, reduzindo assim o volume de

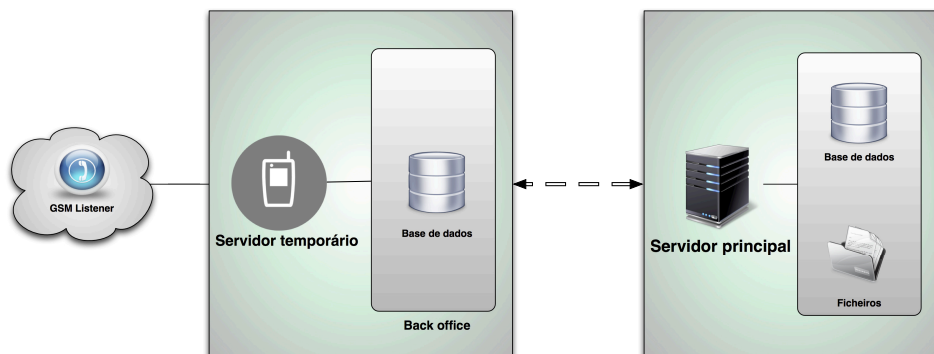


Figura 3.3: Arquitectura do Servidor Temporário

tráfego.

Este servidor possui algumas limitações relativamente ao servidor principal, sendo apenas possível a troca de posições via GSM entre os clientes e o servidor. Não foi desenvolvido o módulo de conexão *web* uma vez que as redes *ad hoc* dos dispositivos móveis possuem fraco alcance e durante uma actividade facilmente se quebraria a ligação entre o cliente e o servidor, levando à perda e a uma inconsistência de dados. Quando terminada a actividade o servidor temporário deverá estabelecer, assim que possível, uma ligação ao servidor principal de modo a enviar os dados referentes à actividade decorrida.

Cliente O modelo de funcionamento do cliente a desenvolver para a aplicação pode ser visualizado na Figura 3.4.

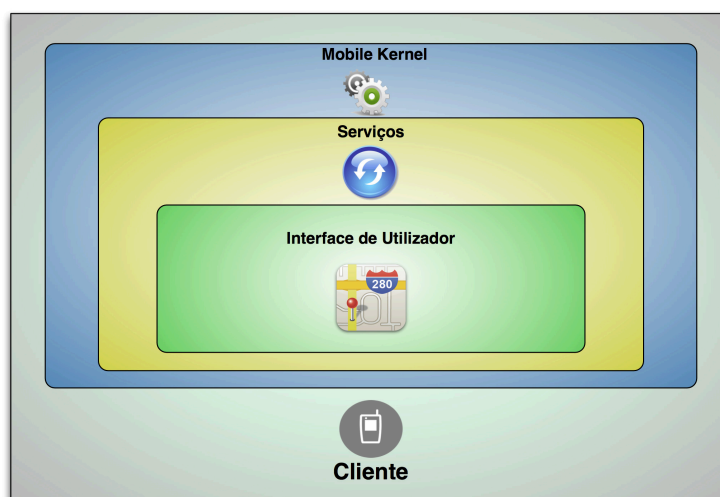


Figura 3.4: Arquitectura do Cliente

A aplicação cliente será assim composta por diversos módulos, sendo a camada representada na imagem por *Interface de Utilizador* aquela que permite acesso e controlo de todas as funcionalidades da aplicação, isto é, será aquilo que é visível para o utilizador da aplicação. Esta poderá possuir algumas diferenças de plataforma para plataforma, devido ao uso das bibliotecas gráficas de cada sistema operativo, no entanto as funcionalidades implementadas serão as mesmas: *login*, registo, visualização de actividades, manipulação de grupos e actividades e sincronização de dados entre o cliente e o servidor. A camada de serviços pode ser explicada juntamente com a camada do *kernel* uma vez que ambas se encontram interligadas. Estas camadas representam as funcionalidades nativas do dispositivo que serão utilizadas e os serviços que correm juntamente com a aplicação, sendo um deles o serviço de envio e recepção de posições que fará uso do serviço de posicionamento do sistema operativo, bem como das conexões *web* ou GSM que são utilizadas para enviar e receber informações. A escrita e leitura de ficheiros também é realizada nestas camadas, bem como o uso do motor de base de dados do dispositivo para o armazenamento de informação.

3.3.2 Identificação de Requisitos

Requisitos Não-Funcionais

A seguinte secção apresenta o conjunto de requisitos não-funcionais necessários à aplicação.

- **Sistema Operativo:** no caso da aplicação pretendida serão analisados dois sistemas operativos para os quais são desenvolvidas aplicações específicas: *Android* e *iOS*. Estes dois sistemas serão aqueles em que se pretende executar aplicações nativas específicas a cada um. A inclusão de um módulo *web* permite também que a aplicação possa ser executada noutros sistemas operativos através de um *web browser*;
- **Interoperabilidade:** outro dos requisitos do sistema é a necessidade de existir uma aplicação cliente para que seja possível executar a aplicação em qualquer plataforma. Para isso será necessário desenvolver um cliente *web* que permita a utilização do projecto para dispositivos que não possuem aplicações nativas;
- **Serviços Externos:** é necessário recorrer ao uso de serviços fornecidos por terceiros, neste caso o serviço de mapas utilizado na aplicação, o serviço de geolocalização e o serviço de GPS para obter a localização específica do utilizador;
- **Configuração de serviços:** na aplicação há a necessidade de existir um conjunto de serviços com os quais os clientes possam interagir, bem como

serviços que permitam aos administradores de sistema uma correcta configuração do servidor. Este conjunto de serviços é implementado num servidor que responsável por toda a aplicação;

- **Escalabilidade:** a aplicação deverá ser facilmente escalável uma vez que o número de utilizadores poderá aumentar repentinamente. Deve também ser considerado o tipo de cliente que usa o serviço uma vez que cada terminal móvel possui um conjunto de características que poderão alterar a usabilidade da aplicação;
- **Segurança:** a aplicação deverá implementar medidas de segurança pois cada utilizador efectuará um registo onde guardará os seus dados pessoais. Também as conexões entre cliente e servidor deverão ser encriptadas de modo a evitar qualquer violação dos dados e garantir sua integridade;
- **Privacidade:** deverá ser tido em conta a privacidade dos utilizadores do projecto uma vez que serão partilhados dados pessoais de cada cliente. Cada cliente deverá possuir a sua conta privada e tornar público apenas aquilo que deseja;
- **Integridade:** deverá ser garantida a integridade de toda a informação gerada e guardada, bem como *logs*, configurações e alertas;

Requisitos Funcionais

De seguida é apresentada uma lista com os requisitos funcionais que são necessários ao bom funcionamento da aplicação.

- **Conta de utilizador:** cada utilizador deverá possuir uma conta com a qual se deve identificar para conseguir utilizar a aplicação. Deverá igualmente possuir uma função de registo para novos utilizadores.
- **Criação de grupos e actividades:** a aplicação deverá permitir aos utilizadores a criação, edição e eliminação de grupos e actividades.
 - Os grupos são criados com o objectivo de agrupar pessoas que pratiquem as mesmas actividades em conjunto, podendo ser criados por qualquer utilizador, de modo a tornar a aplicação mais organizada. Cada um dos grupos pode possuir uma ou mais actividades associadas, sendo que apenas os membros do grupo poderão visualizar as suas actividades. Este grupos poderão ser públicos ou privados, conforme o criador assim o desejar, permitindo assim uma maior privacidade no interior de cada grupo;
 - Cada actividade criada será associada a um determinado grupo em particular. As actividades serão criadas no interior de cada grupo, ocorrendo num determinado período de tempo definido pelo utilizador,

para o qual haverá a troca de posições entre os utilizadores registados nessa actividade. Estas actividades devem poder ser personalizadas pelo utilizador, podendo ser utilizados mapas e rotas personalizadas. Estes mapas e rotas devem ser apenas mostrados se o utilizador assim o desejar uma vez que podem afectar a qualidade de utilização da aplicação.

- **Inscrição e abandono de grupos e actividades:** para além da criação, eliminação e edição de grupos deverá também ser possível aos utilizadores inscreverem-se em grupos e actividades já existentes bem como abandonar os mesmos quando assim o desejarem.
- **Troca de posições:** deverá ser permitida a troca de posições entre o cliente e o servidor no decorrer de uma actividade. O utilizador apenas necessita de enviar a sua posição, e o servidor deverá responder com a posição dos restantes utilizadores juntamente com o seu *delay*.
- **Comunicação web:** o servidor deverá possuir módulos que respondam aos pedidos efectuados pelos clientes em dispositivos móveis. Para isso deverá ser criada uma API que fará a resposta aos diferentes pedidos dos clientes.
- **Comunicação GSM:** para além da comunicação *web* o servidor deverá possuir um módulo GSM que permita a troca de posições através da troca de SMS com os utilizadores durante o decorrer de uma actividade. Este módulo permite a ausência de comunicação *web*, sendo a troca de informações realizada apenas através da troca de SMS.
- **Mapas personalizados:** a aplicação deverá permitir a utilização de mapas personalizados nas suas actividades de modo a enriquecer os locais para quais não existe uma boa cobertura por parte do serviço de mapas. A aplicação aceitará mapas num formato de ficheiro de imagem (*jpg* ou *png* por exemplo) que deverão ser carregados pelo utilizador para a plataforma. Posteriormente estes mapas poderão ser adicionados a uma actividade durante o decorrer da mesma.
- **Rotas personalizadas:** do mesmo modo dos mapas personalizados, a aplicação deverá permitir ao utilizador a criação de rotas personalizadas para posteriormente adicionar a uma actividade. A criação destas rotas pode ser realizada de duas formas: através de desenho no mapa ou através de captura com um dispositivo móvel. Estas rotas serão constituídas por um conjunto de pontos, cada um deles contendo um valor de latitude e longitude de modo a que seja possível a sua definição no mapa. Este conjunto de pontos será unido através de segmentos de recta formando assim uma rota.
 - A criação de rotas através de desenho será realizada sobre um mapa na plataforma web, na qual o utilizador poderá desenhar a rota pretendida

através de uma ferramenta disponibilizada que irá permitir o desenho através de cliques no mapa, definindo assim os pontos da rota. No final do desenho o utilizador nomeará a rota que será armazenada na base de dados e ficará disponível para uso nas actividades que criar.

- A criação de rotas através de um dispositivo móvel será realizada através de um módulo adicional que permita ao utilizador gravar o percurso que está a efectuar quando se movimenta. Esta funcionalidade utilizará o sistema GPS do dispositivo de modo a obter informação da posição do utilizador, efectuando uma captura de posição de acordo com o intervalo de tempo ou espaço que o utilizador desejar. Após o utilizador terminar a gravação da rota esta deverá ser gravada na base de dados do dispositivo e, posteriormente, quando o utilizador desejar, enviada para o servidor principal.
- **Personalização da actividade:** outro dos requisitos que a aplicação deve possuir é a possibilidade do utilizador personalizar a actividade que está a visualizar. Deve ser dada a opção para o utilizador escolher o tipo de conexão realizada com o servidor (*web* ou GSM), o intervalo de tempo de actualização dos dados e se deseja ou não visualizar trilhos e rotas personalizadas. Deverá ainda ser apresentado um menu que forneça apoio ao utilizador para que este consiga decidir mais facilmente qual o tipo de conexão ou intervalo de tempo que deverá utilizar.
- **Multitasking:** a aplicação deverá fazer o uso do mecanismo de *multitasking* para que em casos em que o utilizador necessite de realizar outras tarefas com o dispositivo a aplicação continue em pleno funcionamento em *background* sempre que isto seja possível.
- **Servidor Temporário:** deverá ser permitida a existência de um servidor temporário para as actividades desejadas. Este servidor será implementado num dispositivo móvel que fará parte da actividade que se encontra a decorrer e deve conter a informação sobre a actividade e seus utilizadores igual à que se encontra no servidor principal. No final da actividade deverão também ser enviados todos os dados relativos à actividade para o servidor principal.
- **Sincronização:** como a aplicação poderá ser utilizada quer no dispositivo móvel nativamente, quer num *web browser* é necessário que os utilizadores mantenham os seus dados sincronizados entre os dispositivos. Deverá assim ser incluído um método que permita a sincronização dos dados pessoais do cliente entre as plataformas;
- **Funcionamento *offline*:** nos dispositivos móveis deverá ser possível utilizar a aplicação mesmo quando não existe qualquer ligação de dados. Para isso deverão ser criadas as devidas protecções e avisos ao utilizador das aplicações que poderá realizar quando se encontra em modo *offline*.

- **Menus de ajuda:** deverão também existir textos de ajuda, que expliquem cada um dos menus ao utilizador e as devidas situações, facilitando assim a utilização da aplicação.
- **Definições personalizáveis:** as definições como o servidor de cada aplicação, o tipo de conexão ou os períodos de actualização não deverão ser fixos, podendo o utilizador alterar estes valores sempre que assim o desejar, de modo a dar uma maior liberdade à aplicação.

3.3.3 Especificação das Tecnologias Utilizadas

Servidor

Para a aplicação a desenvolver será necessária a configuração de um servidor *web* que servirá como mecanismo de resposta aos pedidos realizados pelos clientes. Serão de seguida explicados os diversos módulos constituintes do servidor bem como a razão de escolha dos mesmos.

Apache Web Server Sendo um dos objectivos do projecto a implementação de um módulo *web* que permita o acesso à aplicação será necessário a presença de um *web server* a fim de responder a esses pedidos. Depois de analisada a oferta existente no mercado foi escolhido o *Apache 2.2* [6]. Este sistema, actualmente o *web server* mais utilizado do mercado [38], sendo desenvolvido com orientação à segurança, escalabilidade e flexibilidade, torna-o uma opção bastante viável face aos requisitos da aplicação a desenvolver. O facto de ser o *software* mais utilizado do mercado torna-o bem documentado e ao mesmo tempo permite a descoberta de *bugs* e respectiva correcção rapidamente. Em termos de segurança um servidor *Apache* está sempre sujeito a falhas como qualquer outro *software*, no entanto, a sua arquitectura permite que este seja bastante robusto neste campo, sendo apresentadas algumas medidas de segurança extras sugeridas pela equipa de desenvolvimento [7] ou pela utilização de módulo com o protocolo SSL.

O *Apache* é também orientado à escalabilidade, sendo facilmente configurável de acordo com o cenário em que a aplicação se enquadra, o que o torna uma boa opção para o projecto uma vez que é possível ser configurado de acordo com o número de clientes que utilizam a aplicação ou mesmo de acordo com a máquina onde o servidor ficará instalado. Em termos de flexibilidade este servidor está construído de modo a suportar praticamente todas as linguagens *web* mais utilizadas e preparado a ser executado em diferentes sistemas operativos, possibilitando assim que o servidor para o projecto desenvolvido possa ser instalado numa maior diversidade de plataformas e tenha suporte a diferentes tipos de linguagens.

MySQL Outro dos requisitos necessários ao funcionamento do projecto será a existência de um Sistema de Gestão de Base de Dados (SGBD) a fim de manter toda a informação relativa aos utilizadores e à aplicação.

Para a escolha do motor de base de dados foram tidos em conta os seguintes requisitos:

- Segurança dos dados;
- Fiabilidade e Integridade de dados;
- Velocidade de utilização;
- Facilidade de gestão;
- Licença de utilização;

Tendo em conta estes requisitos foi realizada uma primeira triagem aos motores, em que foram escolhidos preferencialmente sistemas *open-source* e sem custos associados, tendo sido seleccionados o *MySQL* e o *PostgreSQL*. O *PostgreSQL* é licenciado pela licença BSD[31], que dá a liberdade ao utilizador de utilizar a *software* como desejar, isto é, o utilizador é livre da utilização da aplicação de SGBD podendo utilizar o seu código fonte ¹ para o desenvolvimento de *software* para venda contendo código *PostgreSQL*.

Por sua vez o *MySQL* é distribuído sobre a licença *General Public License* (GNU) e permite a utilização, cópia, modificação e distribuição do SGBD, sendo no entanto necessário adquirir uma licença comercial se for distribuído *software* que incorpore código *MySQL*, sem necessidade de publicar o código fonte completo da aplicação criada².

Comparativamente à velocidade dos dois motores chegou-se à conclusão, de que ambos apresentam resultados bastante semelhantes entre si. Observando um estudo comparativo dos SGBD's [40] verificamos no entanto que os resultados dos dois motores têm resultados na mesma ordem de valores. Relativamente a licenças comerciais, como o código distribuído será apenas a aplicação cliente, isto é, a aplicação *Android*, o servidor ficará a cargo dos administradores da aplicação sem que haja distribuição de código, sendo assim possível a utilização deste motor de base de dados no projecto a desenvolver.

Mediante o estudo realizado sobre as várias possibilidades de SGBD e mediante a análise feita dos resultados desse estudo, foi escolhido o *MySQL*, devido à familiarização do autor do relatório com o motor em trabalhos realizados anteriormente, como o sistema de gestão de base de dados para suporte do projecto.

PHP Em termos de linguagem de servidor optou-se pela utilização da linguagem PHP. Esta tecnologia será a responsável por efectuar os diversos serviços fornecidos pelo servidor, como a interacção com a base de dados, a manipulação de ficheiros e para a construção da API de resposta a pedidos dos cliente de

¹http://wiki.postgresql.org/wiki/FAQ#What_is_the_license_of_PostgreSQL-3F - acessado a 3 de Novembro de 2011

²<http://www.gnu.org/licenses/fdl.txt> - acessado a 3 de Novembro de 2011

dispositivos móveis. O *Hypertext Preprocessor* (PHP) possui as seguintes características:

- Portabilidade: O PHP na versão 5.3 é suportado por diversos sistemas operativos diferentes, entre os quais *Windows*, *Mac OS X* e *Linux*.
- Interligação com o *MySQL*: o PHP está perfeitamente integrado com o SGBD *MySQL* possuindo *drivers* de integração todos os módulos necessários à interacção com uma base de dados deste tipo;
- Execução de comandos de consola - O PHP disponibiliza um conjunto de comandos de consola que são úteis para verificar os ficheiros adicionados, alterados ou eliminados pela aplicação;
- Geração de gráficos - É também possível a integração de bibliotecas adicionais que permitem a criação de gráficos que facilitam a análise estatística dos dados ³;

A análise da linguagem do servidor passou por duas hipóteses: PHP ou JSP. Apesar de uma análise à performance dos dois tipos de linguagem não ter influenciado a escolha do PHP[44] optou-se pela utilização da linguagem PHP pois o processo de integração com o SGBD escolhido é bastante simples [45], sendo escolhidas para o desenvolvimento de diversas aplicações *web*. Da mesma forma o facto do autor do relatório se encontrar bastante familiarizado com esta linguagem foi um factor que levou à sua escolha.

Depois de analisadas as características citadas acima concluiu-se que o PHP é a linguagem que melhor responde aos requisitos do sistema.

Cliente

Nesta secção serão apresentados os diferentes tipos de aplicação cliente a desenvolver no âmbito deste projecto. Serão brevemente explicados os módulos para cada um dos sistemas operativos móveis propostos bem como o módulo *web* implementado.

Android OS O sistema operativo *Android* é constituído por um *kernel* baseado no sistema operativo UNIX. Para o projecto a desenvolver foi criada uma aplicação nativa na linguagem *Java* que é suportada pelo sistema operativo através de uma *virtual machine* própria criada para o efeito denominada Dalvik [9]. Foram estudados outros tipos de linguagens possíveis de utilização em *Android* pois a performance é um requisito fundamental e, apesar de se ter verificado que existe uma melhor performance quando utilizado código nativo do sistema operativo [42], verificou-se que o desenvolvimento neste tipo de linguagem é de baixo nível permitindo realizar apenas chamadas ao código nativo do

³<http://www.boutell.com/gd/> - acedido a 4 de Novembro

sistema operativo, não permitindo o uso de algumas funções das bibliotecas Android, nomeadamente uma delas que será imprescindível ao projecto, a biblioteca *Google Maps*.

Para o desenvolvimento da aplicação cliente foi utilizada a biblioteca fornecida pela *Google* que permite a utilização da plataforma *Google Maps* em *Android*. Foram ainda realizados testes de performance com uma aplicação utilizando *Open Street Maps* e utilizado um módulo *javascript* no interior da aplicação. Estes tipos de alternativas foram submetidos as testes que serão apresentados posteriormente neste relatório, sendo seleccionado aquele que apresentou uma melhor *performance*.

iOS Para o desenvolvimento de uma aplicação que seja executada no sistema *iOS* foi necessária a definição de alguns requisitos ao seu desenvolvimento, nomeadamente a utilização do sistema operativo *Mac OS X* pois apenas neste sistema é possível executar o SDK disponibilizado pela *Apple*. Foi deste modo utilizado o *software Xcode* fornecido pela *Apple* que disponibiliza todas as ferramentas necessárias ao desenvolvimento. A linguagem utilizada é obrigatoriamente *Objective C* uma vez que é a única opção existente. Relativamente à integração de mapas, o *iOS* utiliza nativamente a plataforma *Google Maps*, pelo que foi a escolhida para ser utilizada no desenvolvimento.

Cliente Web O cliente *web* permite o acesso através de qualquer dispositivo à aplicação sem necessidade de instalação de *software* específico.. A construção da plataforma teve em conta os seguintes requisitos:

- Velocidade de utilização;
- Facilidade de utilização;
- Suporte Multi-Plataforma;
- Suporte para mapas;
- Serviços de geolocalização;

Depois de analisados e estudados estes requisitos [41] foi escolhida a plataforma HTML5 para o desenvolvimento uma vez que, segundo os estudos realizados, verificamos que é suportado pelas plataformas móveis mais utilizadas [22]. Esta linguagem oferece integração com diversos serviços como o serviço de geolocalização que permite obter a posição do utilizador através do *browser*. Apesar de este serviço não ser suportado nativamente pela plataforma *Symbian*, a utilização de um *browser* como o *Opera Mini* permite ao utilizador deste sistema operativo usufruir desta funcionalidade [33]. O suporte desta linguagem à navegação *offline* permite a gravação de dados localmente no dispositivo móvel, permitindo à aplicação o funcionamento mesmo quando ocorrem falhas de rede.

Relativamente ao suporte de mapas ,o serviço da *Google* encontra-se também integrado com esta plataforma [18] com o devido suporte de integração de geocalização pela plataforma utilizada.

Google Polyline Encoder Para o armazenamento das rotas existam duas opções disponíveis: o seu armazenamento em ficheiro, com os pontos que definam a rota criada ou então o armazenamento das rotas criadas num formato específico na base de dados. A primeira opção passou pelo armazenamento das rotas em ficheiro, o que não se torna viável se pensarmos na escalabilidade da aplicação, uma vez que se o número de utilizadores for elevado as escritas e leituras de disco serão bastante intensas, bem como o número de ficheiros abertos o que levaria a um consumo excessivo de recursos, contrariando assim os requisitos da aplicação.

Assim procurou-se um modo de armazenar os dados directamente na base de dados. Uma das soluções seria a utilização de uma base de dados com suporte geo-espacial, uma funcionalidade já incluída no SGBD escolhido para o projecto. Contudo, de modo a reduzir a quantidade de informação e tráfego, os únicos dados que necessitamos de armazenar de cada posição são apenas a latitude e longitude de um ponto, sendo que para uma dada rota apenas é armazenado um conjunto de pontos. Procurou-se assim forma de armazenar estes dados, escolhendo uma funcionalidade disponibilizada pela *Google* nos seus mapas denominada de *Google Polyline Encoder* [19].

Esta ferramenta permite que dado um conjunto de pontos, estes sejam convertidos para um código ASCII, podendo assim ser armazenados no formato *varchar* na base de dados sem a necessidade de utilização de módulos extra da base de dados. Este algoritmo é disponibilizado directamente pela *Google* na API do *Google Maps* pelo que não há necessidade de implementação do algoritmo. Do mesmo modo o algoritmo de descodificação do código está também disponível, devolvendo um *array* com o conjunto de pontos que estão codificados. Em termos do desenvolvimento dos dispositivos móveis existem também versões deste algoritmo nas linguagens a utilizar pelo que não será necessária a sua implementação de raiz, apenas o uso dos devidos métodos. A utilização deste algoritmo vem assim manter a simplicidade dos dados na aplicação, reduzindo o espaço necessário, bem como o tráfego entre o cliente e o servidor.

Capítulo 4

Abordagem

No presente capítulo será apresentada a abordagem efectuada relativamente ao desenvolvimento dos módulos e respectivos testes da aplicação. Serão tidos em conta os requisitos apresentados no Capítulo anterior (3) bem como os cenários de funcionamento da aplicação e sua arquitectura.

4.1 Metodologia de Desenvolvimento

Para a realização do desenvolvimento da aplicação optou-se pela divisão da aplicação em diversos módulos, focando o desenvolvimento em cada um desses módulos, avançando para o módulo seguinte apenas quando o módulo a desenvolver se encontrasse devidamente testado e completamente funcional.

Foi assim utilizada uma metodologia ágil com base nos requisitos e tecnologias anteriormente estudados. Os módulos a desenvolver foram planificados, tentando reduzir os mesmos ao espaço de desenvolvimento planeado conforme a complexidade de cada módulo, efectuando posteriormente os testes de modo a garantir o seu funcionamento antes de prosseguir para o módulo seguinte. Estes testes eram efectuados pelo autor do relatório e também pelo orientador do estágio, que realizava igualmente testes de modo a garantir a qualidade dos módulos desenvolvidos.

Em termos de divisão de módulos optou-se pela divisão pelos diferentes sistemas operativos a utilizar (*Android* e *iOS*) e pelos módulos contidos no servidor *web* que são os responsáveis pela resposta aos pedidos dos clientes. O *Android* foi a primeira plataforma escolhida para o desenvolvimento, juntamente com a plataforma *web* pois alguns dos módulos requerem a interacção com esta plataforma.

Devido a contratempos surgidos durante a execução do projecto não foi possível o desenvolvimento para a plataforma *iOS*, no entanto, todo o desenvolvimento foi realizado de forma a futuramente acolher novas plataformas nativas para o projecto. O facto de existir uma grande fragmentação do sistema *Android*, isto é, a diferente quantidade de dispositivos com diferentes características foi outro dos

motivos que atrasou o desenvolvimento desta plataforma uma vez este factor levou a aumentar o número de testes de compatibilidade e a melhoria dos métodos desenvolvidos para que pudessem ser suportados por sistemas com menor capacidade de processamento.

É ainda de referir que o projecto foi desenvolvido utilizando as últimas versões de cada uma das tecnologias utilizadas, em particular para o sistema Android, o qual foi utilizado para o desenvolvimento o novo sistema 4.1 (*Jelly Bean*) que apesar de não estar ainda disponível no mercado, já se encontrava disponível para *developers* quando terminado este relatório, sendo também todos os módulos igualmente testados no sistema 2.3 (*Gingerbread*) que se trata da versão *Android* mais utilizada de momento do mercado [1].

4.2 Descrição Geral da Arquitectura

A Figura 4.1 representa o diagrama geral da arquitectura mediante a implementação, no qual se apresentam os módulos desenvolvidos e o respectivo fluxo de informação.

Sendo as actividades e partilha de posições o ponto forte de toda a aplicação a desenvolver foi concentrado o desenvolvimento numa partilha o mais simples e eficaz possível. O diagrama representa todo o fluxo de informação trocado entre os clientes e o servidor durante o decorrer de uma actividade.

Ao analisarmos o cliente, verificamos que cada um possui um ficheiro de configuração, onde é indicado qual o endereço do servidor *web* e GSM, bem como o período com que os dados deverão ser actualizados. Estes parâmetros são definidos por omissão durante a primeira instalação da aplicação, sendo este ficheiro criado no dispositivo do utilizador e lido cada vez que a aplicação é executada. Após a primeira execução o utilizador é livre de modificar estes parâmetros sempre que o desejar, sendo o ficheiro de configuração imediatamente gravado após cada modificação de modo a evitar a perda de dados.

Estando os parâmetros de configuração carregados em memória começa então o processo demonstrado na figura 4.1. O parâmetro *actualiza posição* representa uma *thread* que executa o envio automático das coordenadas actuais do cliente, através de uma chamada ao *location listener* do dispositivo e, posteriormente, esta informação é enviada de acordo com o protocolo que o utilizador seleccionou (*web* ou GSM) para o servidor.

Do lado do servidor, ao serem recebidas estas informações são gravadas na base de dados do sistema. De seguida é realizado um novo pedido à API do servidor que envia as posições dos restantes utilizadores da actividade para o cliente, conforme o protocolo pelo qual recebeu o pedido. Ao receber a informação vinda da parte do servidor o cliente apenas trata de correr um método que desenha no mapa os restantes utilizadores da actividade nas posições correspondentes.

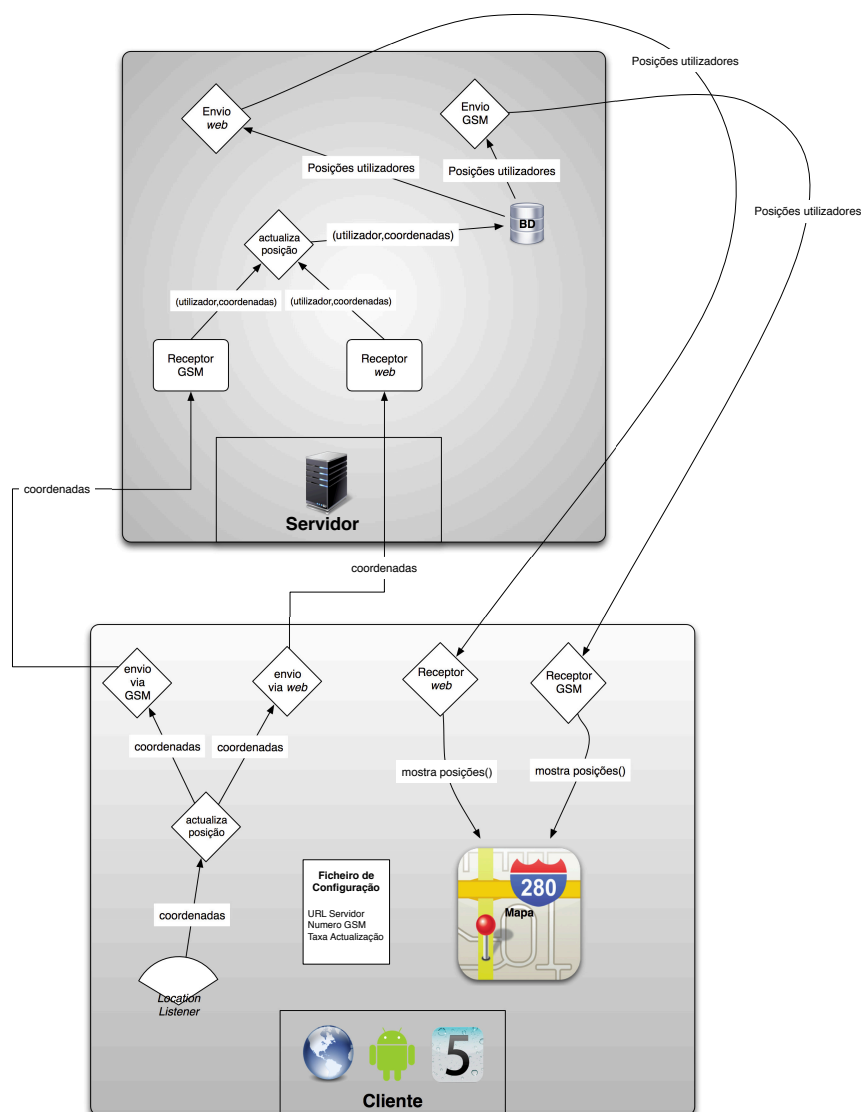


Figura 4.1: Fluxo Servidor - Cliente

4.3 Base de dados

Como já referido na secção de requisitos (3.3.3) houve necessidade da construção de uma base de dados para armazenar os dados referentes à aplicação a desenvolver. Esta base de dados é constituída por seis tabelas, sendo representados na Figura 4.2 os seus atributos e relacionamentos.

Serão de seguida explicadas as tabelas mais relevantes. A tabela *USERS* é a responsável por guardar os dados pessoais referentes a cada um dos utilizadores e é constituída pelos seguintes atributos:

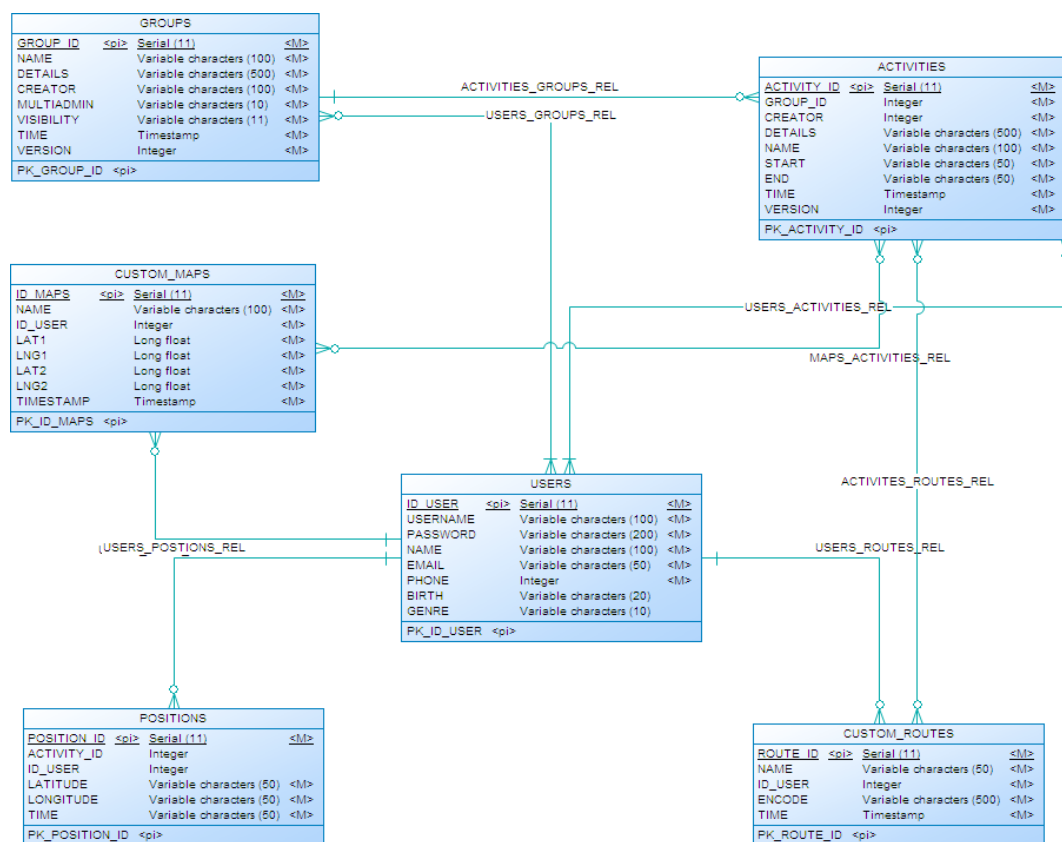


Figura 4.2: Diagrama da Base de Dados

- *id_user*: chave primária, identificador único de cada utilizador;
- *username*: *username* do utilizador, deverá ser único pois será o identificador utilizado para realizar o *login* na aplicação;
- *password*: juntamente com o *username* serve para realizar a autenticação do utilizador;
- *name*: nome completo do utilizador;
- *e-mail*: endereço de *e-mail* do utilizador usado pela aplicação para enviar notificações;
- *phone*: número de telefone do utilizador que será utilizado durante as comunicações utilizando o protocolo *gsm*;
- *birth*: data de nascimento do utilizador;
- *genre*: sexo do utilizador;

Por seu lado a tabela *GROUPS* é a responsável por guardar toda a informação de um grupo, possuindo na sua constituição os seguintes atributos:

- *group_id*: chave primária, identificador único de cada grupo;
- *name*: nome do grupo, deverá igualmente ser único para evitar ambiguidades;
- *details*: detalhes do grupo, poderá ser uma breve descrição do grupo por exemplo.
- *creator*: corresponde ao id do utilizador que criou o grupo;
- *multiadmin*: indica se o grupo poderá ser modificado por qualquer utilizador ou apenas pelo criador;
- *visibility*: indica se o grupo é público, isto é, qualquer um pode encontrar o grupo e aderir, ou privado em que apenas os membros podem adicionar novos utilizadores;
- *time*: data e hora de criação do grupo;
- *version*: versão em que se encontra o grupo, sendo incrementada a cada modificação realizada nos parâmetros do grupo;

A tabela *ACTIVITIES* representa toda a informação relativa às actividades existentes, sendo constituída da seguinte forma:

- *activity_id*: chave primária e identificador único de cada actividade;
- *group_id*: grupo ao qual está associada a actividade;
- *creator*: nome do utilizador que criou a actividade;
- *details*: detalhes a actividade;
- *name*: nome da actividade;
- *start*: data e hora em que se inicia a actividade;
- *end*: data e hora em que termina a actividade;
- *time*: data e hora de criação da actividade;
- *version*: versão em que se encontra o grupo, sendo incrementada a cada modificação realizada nos parâmetros do grupo;

Quanto à tabela *POSITIONS* esta representa as últimas posições dos utilizadores nas actividades, guardando para isso o respectivo identificador do utilizador e da actividade, os dados da posição relativos à latitude e longitude e o instante em que essa posição foi actualizada. De notar que esta tabela apenas regista a posição mais recente de um utilizador numa determinada actividade, isto é, o mesmo utilizador poderá ter várias entradas nestas tabela, no entanto apenas uma por cada actividade em que se encontra inscrito.

A tabela *CUSTOM_MAPS* refere-se a toda a informação relativa aos mapas personalizados, sendo necessário para cada um o nome que corresponde ao nome do ficheiro do mapa, o nome do utilizador que criou o mapa e referências de latitude e longitude que indicam a posição cartesiana do mapa. Para estas referências são pedidas, em coordenadas cartesianas, a posição onde desenhar o canto inferior esquerdo do mapa (*lat1* e *lng1*) e o canto superior direito (*lat2* e *lng2*) de modo a conseguir calcular todos os pontos onde deverá ser renderizado o mapa personalizado. Tal como nas tabelas anteriores é também criado um parâmetro denominado *timestamp* que define a data e hora de criação do mapa.

De modo semelhante a tabela *CUSTOM_ROUTES* define as rotas personalizadas, sendo necessário apenas guardar o nome da rota e do utilizador que a criou, e a própria rota no formato *Encoded Polyline*. Tal como nas tabelas anteriores é também criado um parâmetro denominado *timestamp* que define a data e hora de criação do mapa.

4.4 Módulos Implementados

Como já havia sido referido anteriormente na Secção 4.1 optou-se pela divisão do trabalho em pequenos módulos, pelo que cada um deles realizará uma função específica descrita nos requisitos. Serão de seguida explicados cada um dos módulos desenvolvidos, sendo primeiramente apresentados aqueles que são globais dos dois tipos de aplicação (*web* e *Android*) e posteriormente os módulos específicos de cada uma das partes. Seguidamente, na secção 4.5 serão apresentados os testes realizados sobre esses mesmo módulos e sobre a aplicação final.

4.4.1 Visualização de Mapas

Neste módulo foram desenvolvidos diferentes tipos de visualização de mapas em *Android* com o objectivo de verificar qual o mais adequado ao projecto a desenvolver. Estes módulos servirão apenas para definir qual o tipo de mapas a utilizar, não sendo referidos na secção de testes. Para cada um dos mapas foi implementada uma função de *zoom* e aplicação de *overlays* de modo a testar o seu comportamento.

Foram desenvolvidos três tipos de cenários diferentes:

- Mapas nativos *Android*;

- Mapas utilizando bibliotecas *Open Street Maps*;
- Mapas web utilizando *javascript*;

No desenvolvimento nativo de mapas foi utilizada a biblioteca da *Google* que faz uso dos mapas *Google Maps*, contendo toda a informação disponibilizada na plataforma *online*, sendo as bibliotecas utilizadas parte do sistema operativo pelo que não foi necessário adicionar qualquer código extra. O uso de cache e do mecanismo de mapas *offline* [5] permitiu também a utilização da aplicação sem necessidade de ligação de dados.

Para o desenvolvimento *Open Street Maps* foi necessário a inclusão de algumas bibliotecas adicionais ao projecto, verificando-se um tempo de resposta maior por parte do dispositivo. Outro dos problemas desta *framework* é a fraca cobertura geográfica que possui, em que algumas das áreas visualizadas não continham qualquer informação, surgindo um mapa em branco.

Por último foi desenvolvida uma aplicação que continha uma *web frame*, que apresentava directamente o mapa no dispositivo como se se tratasse de um mapa *Google Maps* visualizado num *browser*. Esta aplicação apresentou diversos problemas no campo de *performance* pois a renderização do *javascript* congelava por certos períodos de tempo o dispositivo, causando até por vezes o *crash* da aplicação.

De seguida é apresentada uma Tabela 4.1 de comparação entre as três plataformas, sendo classificados entre bom (B), mau (M) e não disponível (N/D):

	<i>Google Maps</i>	<i>Open Street Maps</i>	<i>Javascript Maps</i>
Bibliotecas adicionais	B	M	B
Estabilidade	B	B	M
Área geográfica abrangida	B	M	B
Documentação	B	M	B
Funcionalidades	B	M	B
Cache	B	B	M

Tabela 4.1: Tabela de comparação entre plataformas de mapas

Com uma análise à Tabela 4.1 verificamos que a aplicação nativa presente na plataforma *Android* consegue bons resultados em todos os campos analisados. Foi atribuída a classificação de "Bom" ao campo de bibliotecas adicionais devido ao facto de não necessitar das mesmas, é bastante estável uma vez que é desenvolvida especificamente para o sistema operativo em questão, encontra-se bastante bem documentada e por fim permite ainda guardar os mapas para uma posterior utilização sem necessidade de uma ligação de dados.

Quanto à aplicação criada na plataforma OSM verificamos que esta necessita de bibliotecas adicionais à sua utilização, perdendo neste campos para as restantes plataformas testadas. Como já referido anteriormente a área geográfica abrangida por esta plataforma também não é tão boa como na plataforma *Google*, encontrando-se com pouca informação em algumas áreas afastadas dos grandes centros urbanos que ainda não possuem qualquer cobertura geográfica, o que para o projecto a desenvolver, em que pode ser praticado, por exemplo, uma actividade desportiva numa floresta, não se torna viável a falta de cobertura geográfica nesses locais. Por fim em termos de documentação esta também não se encontrava muito desenvolvida, os textos de apoio encontrados na *internet* eram escassos e as funcionalidades disponíveis menos completas comparativamente à *framework* da *Google*. Por fim em termos de *cache* este serviço está bastante optimizado sendo possível, por exemplo, descarregar um mapa completo de um determinado local ou mesmo país no formato comprimido, para posterior utilização sem a necessidade de uma ligação de dados.

Por fim a plataforma *Javascript Maps* encontra-se em termos de documentação e funcionalidades bastante semelhante à aplicação nativa *Android*, no entanto, devido a ser *web based* apresenta alguns problemas: a sua estabilidade é bastante instável, consumindo muitos recursos ao dispositivo, deixando muitas vezes de responder aos comandos do utilizador. Como se trata de uma aplicação *web* também o uso de *cache* também não é controlado, sendo perdidas a maioria as informações descarregadas quando é fechada a janela do mapa.

Depois de analisados todos os parâmetros decidiu-se assim usar os mapas *Google* nativos da plataforma *Android* para desenvolvimento do projecto.

Para a plataforma *web* criada decidiu-se também utilizar *Google Maps* para o seu desenvolvimento, sendo assim utilizadas as mesmas tecnologias em ambas as plataformas.

4.4.2 Mapas Personalizados

Neste módulo pretendia-se implementar um dos requisitos do projecto que se tratava da submissão e visualização de mapas personalizados. A funcionalidade para o envio de mapas sob a forma de uma imagem foi apenas criada na plataforma *web* pois devido à dimensão reduzida dos ecrãs dos dispositivos móveis não faz sentido desenvolver esta opção para os mesmos.

Criou-se assim uma página HTML na qual o utilizador selecciona o ficheiro pretendido e coloca as coordenadas cartesianas do canto inferior esquerdo e superior direito onde deverá ser colocado o novo mapa personalizado. Este procedimento guarda no servidor a imagem, numa directoria específica para o utilizador em questão e também os seus dados na base de dados, sendo toda a manipulação de ficheiros realizada pelo PHP.

Em termos de submissão do mapa na plataforma *web* este é feito através de uma ferramenta disponibilizada na API do *Google Maps* denominada *GroundOverlay* que, indicando os pontos cartesianos do mapa personalizado anteriormente

referidos, é possível renderizar sobre o mapa da Google a imagem desejada. Esta imagem sofre também ajustes de tamanho quando são realizadas operações de *zoom* sobre o mapa original.

Quanto à plataforma *Android* foram encontradas algumas dificuldades na aplicação dos mapas personalizados. Apesar de ser possível realizar o posicionamento de *overlays* que são automaticamente redimensionados com o uso de *zoom* no mapa na plataforma *web* do *Google Maps* esta função não se encontra disponível nas bibliotecas dos mapas para *Android*. Deste modo foi necessário contornar esta situação utilizando o uso normal de *overlays* disponível na plataforma móvel, que permitem o uso de imagens que são automaticamente convertidas e apresentadas no mapa. O problema encontrado com estas imagens foram as suas constantes proporções, isto é, assim que a imagem é convertida e aplicada no mapa esta mantém o seu tamanho constante, independentemente do nível de *zoom* aplicado ao mapa. Para resolver esta situação criou-se um método que ao ser realizado qualquer evento de *zoom* num mapa, calcula as dimensões da imagem a aplicar de acordo com um rácio previamente definido, sendo possível assim a realização de *zoom* em ambos os componentes. Devido à grande fragmentação que o sistema operativo *Android* possui, isto é, o número de diferentes dispositivos com diferentes capacidades de processamento que correm esta sistema operativo, foi ainda criado um método que comprime as imagens dos mapas personalizados antes de serem aplicadas no mapa original. Foi também colocado como opcional o carregamento dos mapas, isto é, mesmo que a actividade a visualizar contenha mapas, o utilizador é livre de não os carregar para não afectar a *performance* do seu dispositivo.

4.4.3 Criação e visualização de Rotas

A criação deste módulo foi dividida em duas partes: criação de rotas e visualização das mesmas.

Para a criação de rotas com o dispositivo *Android* foi utilizado apenas o sinal GPS do dispositivo, não sendo permitida a sua criação enquanto não fosse adquirido este sinal, de modo a obter uma rota o mais precisa possível. Foram também adicionadas opções para que possam ser alterados os valores de captura de posição, em segundos e metros. O módulo realiza a captura de posições conforme o período desejado, sendo a rota que está a ser capturada desenhada no ecrã sobre um mapa, sempre que são capturadas novas coordenadas. No final é pedido ao utilizador para nomear a rota e esta é guardada automaticamente na base de dados do dispositivo, no formato *Encoded Polyline*.

Do lado da plataforma *web* optou-se por uma abordagem diferente: como os dispositivos que acedem a esta plataforma são maioritariamente computadores, dificilmente haveria possibilidade de captura de uma rota, uma vez que a sua utilização é fixa. Assim desenvolveu-se para a plataforma *web* uma ferramenta

de criação de rotas via cliques sobre o mapa, em que cada um dos cliques define um ponto da rota a criar. Assim é mostrado ao utilizador um mapa limpo onde este pode desenhar a rota que pretende onde desejar. Tal como na plataforma *Android*, no final o utilizador atribui um nome à rota criada e esta é armazenada directamente na base de dados principal do sistema já sob o formato *Encoded Polyline*. Esta opção foi pensada para ecrãs de grande dimensão, não sendo viável a sua utilização em ecrãs de pequena dimensão como *smartphones*, daí o não desenvolvimento deste tipo de criação de rotas para a plataforma *Android*.

A visualização de rotas é feito de modo semelhante nas duas plataformas. Ao ser iniciada uma actividade, caso esta contenha rotas personalizadas, é carregado o código de cada uma das rotas e de seguida, convertido para um *array* de pontos que posteriormente é desenhado e mantido sobre o mapa durante toda a actividade. Tal como nos mapas personalizados este carregamento é opcional por questões de performance do dispositivo que utiliza a aplicação.

4.4.4 *Login e Registo*

Foi criada para ambas as plataformas um módulo que permite um registo de um novo utilizador na aplicação bem como o *login* de um utilizador já existente. No registo são pedidas todas as informações necessárias sobre o utilizador como o *e-mail*, número de telemóvel e *username*, garantindo que são únicos para cada utilizador e ao mesmo tempo garantindo que os dados são introduzidos no formato correcto de forma a não causar problemas de comunicação no futuro. O utilizador é notificado do seu registo através da recepção de um *e-mail* na caixa de correio electrónico indicada.

No caso da plataforma *Android* foi ainda utilizado um sistema que permite guardar os dados de *login* do utilizador, evitando assim que seja necessário introduzir o nome de utilizador e a palavra-passe de cada vez que o utilizador abrir a aplicação. Este dados são assim guardados numa zona do sistema operativo *Android* denominada *SharedPreferences* desenhada para armazenar este tipo de dados. Como estes dados, apesar de não se encontrarem acessíveis ao utilizador normal, estão sempre sujeitos a ser violados por um *hacker*. Dado isto, optou-se por criar um sistema de encriptação dos mesmos antes do seu armazenamento.

A edição de perfil está apenas disponível na aplicação *web*.

4.4.5 *Operações de Grupos e Actividades*

Foram criadas diversas operações sobre grupos e actividades. A plataforma *web* é a mais completa pelo que será a primeira a ser explicada.

É dada a possibilidade ao utilizador de criar um novo grupo, publico ou privado, ficando automaticamente inserido no mesmo. É possível também a edição dos dados do grupo, caso o utilizador seja o criador desse mesmo grupo. Por

fim é dada a possibilidade de eliminação definitiva do grupo, sendo o utilizador alertado para o facto das actividades associadas a esse grupo serem igualmente eliminadas. É também dada a possibilidade ao utilizador de se juntar a grupos já existentes, bem como o abandono dos mesmos, sendo também nesta situação notificado que ao abandonar um grupo abandonará todas as actividades do mesmo. Em termos de actividades é também dada ao utilizador a possibilidade de criar novas actividades, tendo para isso de escolher um grupo ao qual pertença para adicionar essa mesma actividade. Durante a criação devem ser definidas a data e hora de inicio e fim da mesma. Tal como nos grupos é possível para o utilizador que criou a actividade editar a mesma bem como proceder à sua eliminação. A possibilidade de se juntar a novas actividades também está disponível, mas apenas para actividades que se encontrem dentro dos grupos nos quais o utilizador se encontre inscrito, sendo também dada a possibilidade de abandono de uma actividade. Por fim existe ainda a possibilidade de associar um mapa personalizado ou uma rota personalizada a um mapa, desde que o utilizador tenha criado esses mesmos extras.

Para a plataforma *Android*, por uma questão de simplicidade apenas foi dada a possibilidade de junção ou abandono de grupos e actividades. É necessário uma ligação de dados para a junção a novos grupos e actividades, bem como para o seu abandono, de modo a que os dados sejam actualizados no servidor principal e visíveis para os restantes membros.

4.4.6 Visualização de actividades

A visualização de actividades decorre de modo semelhante nas duas plataformas.

Na plataforma *web* é dado ao utilizador a lista de actividades que se encontram a decorrer no momento. Depois de seleccionar a actividade é apresentado um mapa com os dados actuais da actividade. Inicialmente, caso existam associados à actividades, não são carregados mapas ou rotas personalizadas, sendo o utilizador livre de os activar e desactivar através de uma simples *checkbox* lateral. São também mostrados botões que permitem a actualização da posição bem como a actualização do mapa de modo a actualizar a posição dos restantes utilizadores da aplicação. De referir que como todas as operações são realizadas sobre o *javascript* do *Google Maps* não havendo necessidade de actualizar a página que se encontra aberta. A posição do utilizador é adquirida através da geolocalização disponível no HTML5.

Para a plataforma *Android* é igualmente dado ao utilizador a possibilidade de escolher a actividade a visualizar numa lista de actividades a decorrer qual a que deseja visualizar. É ainda possível escolher anteriormente a ser carregado o mapa original se deseja mapas e rotas personalizados (desactivados por omissão), o tipo de conexão a efectuar (*web* ou GSM) e a alteração dos valores de actualização

do mapa. Depois de carregado o mapa este é actualizado conforme o período escolhido pelo utilizador, sendo o utilizador alertado da mesma actualização. O envio foi implementado criando uma classe que apenas faz a chamada aos *listeners* por um pequeno período de tempo quando é necessária a actualização da posição, sendo assim poupada bastante energia pois o sistema GPS não se encontra constantemente em funcionamento. Por fim são ainda fornecidos dois botões dentro do menu do mapa que permitem que a operação de envio de posição e recepção sejam feitos de forma independente e manual.

Se a troca de coordenadas for realizada pelo protocolo GSM o dispositivo trocará as informações através de SMS's, estando o dispositivo a receber SMS a cada actualização. Foi criado um *token* de identificação específico da aplicação que permite distinguir este tipo de SMS dos restantes recebidos pelo dispositivo, sendo apenas recolhidas informações dos SMS's recebidos exclusivamente para a aplicação em causa.

4.4.7 Sincronização

Para permitir o funcionamento *offline* da aplicação foi necessário a criação de um módulo que permitisse a sincronização de dados entre o cliente e o servidor. Embora a aplicação não possa ser utilizada totalmente *offline* (pois é necessária a troca de informações durante uma actividade) deverá ser possível visualizar os dados de cada grupo e actividade sem a necessidade constante de ligação ao servidor. Assim criou-se um método que cria uma base de dados local no dispositivo móvel, com as informações que o utilizador possui no servidor principal. Esta sincronização está disponível através de um botão colocado no menu principal da aplicação, sendo sempre realizada de forma manual pelo utilizador de forma a poupar o consumo de recursos e o volume de tráfego utilizado pela aplicação quando não desejado. Quando ocorre uma sincronização apenas os dados referentes ao utilizador são descarregados, isto é, os seus grupos e actividades e respectivos mapas e rotas personalizadas, caso ainda não se encontrem no dispositivo. É mantido um formato idêntico ao da base de dados principal do sistema, sendo a sincronização controlada pelo campo *version* nas tabelas da base de dados. Dados como rotas personalizadas criadas no dispositivo são também enviados neste momento, caso não exista ligação de dados aquando da criação da mesma.

Do lado da plataforma *web* não faz sentido existir este tipo de mecanismo pois a própria interface é apresentada pelo servidor da aplicação, sendo necessária uma ligação de dados constante.

4.4.8 API

De forma a responder aos pedidos dos clientes móveis foi criada uma API REST em PHP, com todos os ficheiros necessários a cada um dos pedidos. Esta

API foi desenvolvida não a pensar somente na plataforma *Android* mas a qualquer plataforma móvel para qual o projecto venha a ser desenvolvido. Cada um dos ficheiros está preparado para receber por parâmetro ou *HTTP Post* os dados do cliente, devolvendo a resposta ao pedido no formato JSON. A API está presente no servidor principal da aplicação para que os dados sejam coerentes entre as duas plataformas.

Na plataforma *Android* foi criada uma classe onde é possível alterar aos programadores qual o URL do servidor onde se encontra presente a API, funcionando todas as restantes classes de sincronização com base nesta configuração.

4.4.9 Servidor

A linguagem utilizada do lado do servidor foi o PHP e do lado do cliente foi necessário a implementação de *javascript* para a protecção de alguns campos e do uso de mapas, uma vez que o *Google Maps* funciona sobre esta linguagem. Foi também utilizada a biblioteca *jQuery* [24] que permite a utilização de diversas funções previamente definidas na linguagem *javascript*. A manipulação de ficheiros e da base de dados é toda ela realizada pelo PHP, utilizando as bibliotecas já incluídas nesta plataforma sem a necessidade da instalação de *software* adicional.

A base de dados está preparada para correr de forma independente noutro servidor, se necessário, tendo sido criado os devidos ficheiros de configuração para o efeito.

Em termos de criação de um servidor secundário *Android* foi criada uma aplicação simples, que permitisse a troca de SMS entre o dispositivo e os clientes. Por uma questão de escalabilidade não são armazenadas as rotas dos utilizadores durante a actividade, logo os únicos dados trocados são as posições via SMS. Este servidor utiliza um mecanismo de sincronização semelhante, sendo necessário um *login* de um dos utilizadores da actividade para que irá ser utilizado, evitando assim a cópia integral de toda a informação do servidor principal. Posteriormente a aplicação encontra-se sempre à escuta de novos SMS's que cheguem ao dispositivo, comparando estes com chave identificadora da aplicação. Se se tratar de uma SMS específica da aplicação, o servidor armazena os dados desse utilizador, e responde para o mesmo número com os dados dos restantes utilizadores. De notar que este servidor é bastante limitado pois o texto da mensagem escrita poderá ficar excessivamente grande com um elevado número de utilizadores.

4.5 Carga de Testes

Esta secção descreve a carga de testes efectuada aos mecanismos apresentados no capítulo anterior, tendo em conta o requisitos e o planeamento definidos.

4.5.1 Cenário de Testes

Para a realização dos testes ao trabalho desenvolvido foi necessária a instalação das tecnologias num servidor externo e em dispositivos *Android*. Para os testes durante o desenvolvimento dos módulos foi utilizado apenas um dispositivo móvel fornecido pelo DEI. As características de cada um podem ser visualizadas nas Tabelas 4.5.1 e 4.5.1.

Processor Name	AMD Athlon(tm) XP 2000+
Processor Speed	1.6 GHz
Total Number Of Cores	1
L2 Cache	512 KB
Memory	1 GB DDR
System Version	Ubuntu Server 12.04.1 LTS x84
Kernel Version	3.2.0-29
Apache Version	2.2.22
MySQL Version	5.5.24
PHP Version	5.3.10-1

Tabela 4.2: Especificações do Servidor

Model	Samsung Galaxy Mini S5570
CPU	600 MHz ARMv6
GPU	Adreno 200
Display	240 x 320 pixels, 3.14 inches
Memory	384 MB
System Version	Android OS, v2.3.5 (Gingerbread)
Kernel Version	2.6.35.7

Tabela 4.3: Especificações do dispositivo *Android*

Estes testes foram realizados após o desenvolvimento de cada módulo, verificando se estes cumpriam todos os requisitos necessários. Para a análise dos testes foram utilizados os requisitos definidos na secção 3.3.2, comparando os resultados obtidos de forma a verificar se estavam em conformidade com os requisitos. Além dos testes pelo autor do relatório todos módulos foram igualmente testados posteriormente pelo orientador do projecto de modo a garantir a sua qualidade e fiabilidade e discutidos com o autor do relatório de forma a corrigir possíveis erros e melhorar funcionalidades. Os testes finais, após a junção de todos os módulos, foram realizados com um grupo de 4 dispositivos *Android*, juntamente com um cliente *web*.

4.5.2 Testes Não-Funcionais

Após a conclusão dos testes realizou-se um balanço para garantir que os requisitos não-funcionais (ver 3.3.2) foram cumpridos. A Tabela 4.4 apresenta os requisitos segundo a implementação.

Módulo	Descrição do teste
Sistema Operativo	Para a aplicação cliente foi escolhido o sistema operativo <i>Android</i> que integra todas as tecnologias necessárias ao desenvolvimento da aplicação. O servidor escolhido corre o sistema operativo <i>Ubuntu</i> que sendo um sistema <i>Linux</i> integra na perfeição a plataforma LAMP escolhida para o projecto.
Interoperabilidade	A criação de uma plataforma <i>web</i> com todas as funcionalidades presentes nos requisitos do projecto permitem que este seja executado por qualquer plataforma.
Serviços Externos	O serviço de mapas da <i>Google</i> fornece toda a documentação necessária bem como ferramentas necessárias ao desenvolvimento do projecto.
Escalabilidade	Através dos estudos efectuados anteriormente verificamos que o sistema implementado para o servidor LAMP satisfaz este requisito para um número razoável de clientes.
Segurança	O uso do sistema de <i>login</i> com o mecanismo de <i>cookies</i> garante segurança à aplicação, assim como a conexão à API móvel através de conteúdo encriptado.
Privacidade	O mecanismo de expiração de <i>cookies</i> e a possibilidade de criar dados públicos ou privados conferem ao utilizador a privacidade dos seus dados durante a utilização da aplicação.
Integridade	Todos os dados referentes a alertas e tratamento dos mesmos estão devidamente salvaguardados na base de dados. Toda a informação resultante dos mecanismos implementados estão salvaguardados em ficheiros de <i>log</i> próprios.

Tabela 4.4: Requisitos não funcionais segundo a implementação

4.5.3 Testes Funcionais

Após a conclusão dos módulos descritos na Secção 4.4 foram comparados com os módulos funcionais descritos na Secção 3.3.2, verificando quais os requisitos funcionais já satisfeitos. Os resultados são mostrados na Tabela 4.5

Módulo	Descrição do teste
Conta de utilizador	Através do módulo <i>web</i> criado é possível o registo na aplicação e o posterior <i>login</i> , quer na plataforma <i>web</i> , quer em <i>Android</i> . Os campos de inserção também se encontram devidamente protegidos, garantindo assim que os dados são introduzidos no formato correcto. Na plataforma <i>Android</i> foi também possível utilizar a aplicação depois de efectuar o primeiro <i>login</i> , mesmo em modo <i>offline</i> uma vez que os dados se encontram guardados e encriptados nas preferências do sistema.
Criação e eliminação de grupos e actividades	Na plataforma <i>web</i> é possível a realização destas duas funcionalidades, encontrando-se completamente funcionais.
Inscrição e abandono de grupos e actividades	Este requisito foi desenvolvido para ambas as plataformas. A aplicação não permite que um criador de um grupo ou actividade o abandone, pois iria causar uma inconsistência de dados na base de dados do sistema.
Comunicação web	A comunicação <i>web</i> com o servidor foi testada em todos os módulos desenvolvidos através da comunicação entre o dispositivo <i>Android</i> e o servidor. Foi obtida uma resposta positiva em todos os módulos criados, e uma notificação quando o servidor não se encontrava disponível.
Comunicação GSM	Foi testada a comunicação GSM com um servidor <i>Android</i> . A comunicação foi realizada com sucesso, no entanto notou-se um pouco de atraso na resposta relativamente à comunicação <i>web</i> com o uso de três dispositivos a realizar troca de informação simultaneamente com este protocolo.

Continua na página seguinte

Tabela 4.5: Requisitos funcionais segundo a implementação

Tabela 4.5 – *Continuada da página anterior*

Módulo	Descrição do teste
Mapas personalizados	Encontra-se funcional para a plataforma <i>web</i> e <i>Android</i> , sendo os mapas carregados em formato imagem, convertidos e renderizados correctamente sobre o <i>Google Map</i> . Para imagens de maior dimensão notou-se alguma dificuldade da parte do dispositivo <i>Android</i> de lidar com estas imagens, congelando por vezes a utilização dos dispositivo por alguns segundos.
Rotas personalizadas	A sua criação é possível em <i>Android</i> e <i>web</i> . Na plataforma <i>Android</i> o utilizador é alertado para a escolha correcta do tempo e espaço para o qual deseja recolher coordenadas, uma vez que quando o número de pontos de uma rota de torna bastante elevado o dispositivo apresenta problemas no processamento desses dados. Também é possível a sua visualização em ambas, sendo codificadas e decodificadas correctamente, independentemente da plataforma onde a operação seja efectuada.
Personalização da Actividade	A personalização da actividade encontra-se em funcionamento, podendo ser adicionadas rotas e mapas criados pelo utilizador à actividade que este desejar. É possível também activar e desactivar a visualização destes componentes quando desejado. Na plataforma <i>Android</i> é ainda possível escolher qual o servidor <i>web</i> ou GSM desejado e também o período de actualização das posições.
<i>Multitasking</i>	Este requisito foi testado na plataforma <i>Android</i> , recebendo uma chamada enquanto se executava a aplicação. Foi também colocada a aplicação em <i>background</i> no dispositivo, enviado um SMS e voltou-se novamente à aplicação, não se verificando qualquer problema no seu funcionamento.

Continua na página seguinte

Tabela 4.5: Requisitos funcionais segundo a implementação

Tabela 4.5 – *Continuada da página anterior*

Módulo	Descrição do teste
Servidor Temporário	Durante os testes de grupo foi utilizado um dispositivo <i>Android</i> que assumiu a função de servidor GSM da aplicação. Neste teste notaram-se alguns problemas da parte do servidor, uma vez que este demorava algum tempo a ler e processar as respostas dos SMS's recebidos. Se o período de actualização não for muito pequeno (inferior a sessenta segundos) e o número de utilizadores relativamente baixo (cinco ou menos) este servidor é aceitável, caso contrário o tempo de resposta aumenta e os SMS's recebidos acumulam-se no dispositivo, levando à inconsistência de dados.
Sincronização	O método de sincronização foi testado com um dispositivo novo a aceder pela primeira vez à aplicação, sincronização após modificações do lado do servidor e sincronização após existirem novas rotas gravadas no dispositivo. Todas estas corresponderam ao que era pretendido. Foi ainda testado o <i>download</i> de diversos ficheiros mapas personalizados, tendo sido todos estes correctamente gravados no dispositivo.
Funcionamento <i>offline</i>	O funcionamento <i>offline</i> foi testado em toda a aplicação de forma a garantir que foram protegidas todas as funcionalidades para a falta de um protocolo de comunicação. Foi assim possível utilizar a aplicação na visualização de actividades, com os mapas gravados em <i>cache</i> ou previamente colocados <i>offline</i> e também a visualização dos dados que estavam gravados na base de dados local do dispositivo.
Menus de ajuda	Os menus de ajuda estão disponíveis por toda a aplicação, ajudando o utilizador sempre que este possua dúvidas. Para este teste foi pedido que um dos utilizadores que participou nos testes utilizasse a aplicação, guiando-se apenas pelo menus de ajuda.

Continua na página seguinte

Tabela 4.5: Requisitos funcionais segundo a implementação

Tabela 4.5 – *Continuada da página anterior*

Módulo	Descrição do teste
Definições Personalizáveis	Por fim foram modificados vários dos parâmetros da aplicação. Foi alterado o endereço do servidor principal, o tempo de actualização e o endereço GSM, não tendo sido encontrados novos problemas com estes valores.

Tabela 4.5: Requisitos funcionais segundo a implementação

De referir ainda que foi ainda realizado um teste final a todo o projecto, com 4 dispositivos *Android* e um dispositivo *web*, neste caso um computador portátil de modo a verificar a integração de todos os módulos e a interoperabilidade entre os diversos dispositivos. Não foram encontrados problemas durante o decorrer da actividade de teste, sendo a partilha de posições realizada entre os cinco clientes tal como era esperado.

Capítulo 5

Reflexões e trabalho futuro

Este capítulo expõe as conclusões da presente dissertação, o trabalho futuro a realizar no âmbito deste projecto e ainda algumas reflexões sobre o desenrolar dos trabalhos.

5.1 Conclusão da dissertação

O mercado dos dispositivos móveis tem vindo a tornar-se uma tecnologia dominante no mundo actual, sendo cada vez mais utilizados no dia a dia para as mais diversas tarefas. Com este trabalho tentou-se assim desenvolver uma aplicação inovadora, diferente das ofertas de mercado existentes, que permitisse não só algo divertido para o utilizador mas também útil. Foi assim desenvolvido o trabalho descrito durante este relatório, baseado numa arquitectura que permitisse futuramente a sua expansão a outros sistemas operativos móveis e à adição de novas funcionalidades. No desenrolar dos trabalhos o autor tentou sempre ao máximo utilizar as últimas versões das tecnologias utilizadas no projecto, garantindo assim uma maior compatibilidade com os sistemas actuais e futuros.

No presente documento é possível ver de forma clara o processo de construção de raiz da aplicação. Este iniciou-se com a pesquisa por ferramentas existentes, não apenas para garantir que se estava a realizar algo inovador mas também para obter ideias sobre as tendências do mercado actual de forma a enriquecer a solução pretendida. De seguida foram definidas quais as metas a atingir com a mesma. Após o desenho da solução, passou-se à implementação da aplicação de acordo com o planeamento efectuado e as tecnologias estudadas. Antes de finalizar o documento foram realizados os testes à aplicação com os requisitos idealizados. Esta é uma tarefa que se reveste de toda a importância, não apenas porque é a forma do autor provar que conseguiu atingir os objectivos a que se propôs mas também demonstra que a ideia que surgiu no seio do SSE conseguiu ser modelada e implementada, conseguindo construir uma aplicação funcional. Dados os resultados dos testes efectuados aos requisitos, o autor conclui que atingiu os

objectivos a que se propôs: a construção de uma aplicação capaz de realizar a partilha de posições de forma automática e personalizada, que funcionasse em diversos sistemas operativos diferentes.

5.2 Trabalho Futuro

Do ponto de vista do autor o desenvolvimento futuro para um trabalho deste tipo passa pelo lançamento de um produto para a *Play Store*, a loja de aplicações *Android*, depois de submetido ainda a alguns testes mais rigorosos, numa fase *beta* por utilizadores convidados. Após o lançamento do produto no mercado de aplicações seria possível então verificar o sucesso do mesmo, que caso se verificasse, um investimento futuro passaria pelo desenvolvimento para outras plataformas, como *iOS* ou *Windows Phone* com o posterior lançamento na loja de aplicações do mesmo. Ao mesmo tempo estes lançamentos permitem a correcção de pequenos *bugs* encontrados pelos utilizadores e também proporcionam um *feedback* dos mesmos de modo a saber quais os problemas ou sugestões ao produto lançado.

A inclusão de novas funcionalidades também é outro investimento futuro numa aplicação deste tipo. De modo a despertar a curiosidade dos utilizadores seria interessante adicionar novas funcionalidades como um histórico de actividades, que permitisse ao utilizador verificar quais os percursos que foram efectuados durante uma actividade. A extensão a outro tipo de dispositivos como leitores de RFID pode também ser outro cenário interessante de incluir na aplicação, em que poderão ser partilhadas localizações mais exactas no interior de um edifício, por exemplo.

As tecnologias estão em constante avanço, tornando o desenvolvimento deste tipo de aplicações um trabalho em constante mutação, adaptando-se às circunstâncias que vão surgindo, como por exemplo, novas versões do sistema operativo que tornam funcionalidades utilizadas actualmente obsoletas. Nunca será um trabalho totalmente finalizado devido ao avanço da tecnologia, porém, a construção do presente e correcta visualização do futuro podem resultar em melhores soluções.

5.3 Reflexão sobre o desenrolar dos trabalhos

Por fim, serve esta secção para apresentar algumas reflexões do autor sobre o trabalho realizado.

Com o projecto no âmbito dos trabalhos da dissertação terminado, conclui-se que, do planeamento efectuado no princípio do ano, fica por cumprir um dos objectivos inicialmente definidos, sendo este o desenvolvimento para a plataforma *iOS*. Porém, como já justificado (capítulo 4) o autor optou por não iniciar o desenvolvimento para esta plataforma devido a contratempos como problemas de

saúde do mesmo que atrasaram o desenvolvimento *Android* e *web*, levando a que o tempo para o desenvolvimento para esta última plataforma fosse encurtado. O facto da fragmentação encontrada no sistema *Android* também foi outro dos factores que atrasou o desenvolvimento do projecto, pois foi necessário mais tempo para garantir uma aplicação que funcionasse em na maioria das plataformas *Android*, que actualmente abrange uma grande variedade de dispositivos, cada um com especificações diferentes que alteram a usabilidade da aplicação. O autor considera que este foi também um dos principais desafios encontrados no projecto, devido à optimização que foi necessária conferir à aplicação para que todas as suas funcionalidades fossem suportadas.

Em termos de experiência o autor ganhou experiência em diversas áreas, destacando o desenvolvimento na linguagem PHP, *javascript* e para a plataforma Android, sendo esta última considerada a mais importante para o autor pois o elevado crescimento deste sistema operativo torna-o um alvo bastante apelativo para um desenvolvimento de aplicações futuras.

Certamente que a mais valia de conhecimento adquirido foi todo o processo de estudo, desenvolvimento e implementação de todo o projecto o que permitiu ao autor adquirir a experiência e conhecimento necessários para a continuidade, não só deste projecto, mas também de futuros projectos.

Bibliografia

- [1] <http://developer.android.com/about/dashboards/index.html>.
- [2] Android. <http://www.android.com/>.
- [3] Android dalvik. <http://source.android.com/tech/dalvik/index.html>.
- [4] Android, ios activations hit record on christmas. again. http://news.cnet.com/8301-1035_3-57348698-94/android-ios-activations-hit-record-on-christmas-again/.
- [5] Android maps offline. <http://google-latlong.blogspot.pt/2012/06/go-offline-with-google-maps-for-android.html>.
- [6] Apache. <http://httpd.apache.org/>.
- [7] Apache security tips. http://httpd.apache.org/docs/2.0/misc/security_tips.html.
- [8] Automotive navigation data. <http://www.and.com/>.
- [9] Dalvik. <http://developer.android.com/guide/basics/what-is-android.html>.
- [10] Facebook. <http://www.facebook.com/>.
- [11] Facebook compra gowalla. <http://mashable.com/2011/12/02/facebook-buys-gowalla-report/>.
- [12] Foursquare. <http://foursquare.com/>.
- [13] Glympse. <http://www.glympse.com/>.
- [14] Google. <http://www.google.com/>.
- [15] Google latitude. <https://www.google.com/latitude/>.
- [16] Google local. <http://googleblog.blogspot.com/2004/09/journey-may-be-reward-but-so-is.html>.
- [17] Google maps. <http://maps.google.com/>.

- [18] Google maps html5 support. <http://code.google.com/apis/maps/documentation/javascript/tutorial.html#HTML5>.
- [19] Google polyline encoder. <https://developers.google.com/maps/documentation/utilities/polylinealgorithm>.
- [20] Gowalla. <http://gowalla.com/>.
- [21] Html5. <http://en.wikipedia.org/wiki/HTML5>.
- [22] Html5 mobile support. <http://mobilehtml5.org/>.
- [23] ios. <http://www.apple.com/pt/ios/>.
- [24] jquery. <http://jquery.com/>.
- [25] jquery mobile. http://en.wikipedia.org/wiki/JQuery_Mobile.
- [26] Life360. <http://www.life360.com/>.
- [27] Market share - mobile os. http://gs.statcounter.com/#mobile_os-ww-monthly-200908-201208.
- [28] Mysql. <http://www.mysql.com/>.
- [29] Nokia. <http://www.nokia.com/>.
- [30] Nokia & microsoft partnership. <http://www.microsoft.com/presspass/press/2011/feb11/02-11partnership.msp>.
- [31] Open source initiative osi - the bsd license. <http://www.opensource.org/licenses/bsd-license.php>.
- [32] Open street maps. <http://www.openstreetmap.org/>.
- [33] Opera mini features. <http://my.opera.com/operamobile/blog/opera-mobile-11-for-android-and-symbian>.
- [34] Play store and app store market share. <http://www.androidauthority.com/google-play-vs-apple-app-store-2012-76566/>.
- [35] Software and systems engineering. <http://cisucpt.dei.uc.pt/groups/show/4>.
- [36] Symbian. <http://symbian.nokia.com/>.
- [37] Twitter. <http://twitter.com/>.
- [38] Web servers market share. <http://news.netcraft.com/archives/2012/03/05/march-2012-web-server-survey.html>.

- [39] Windows phone 7. <http://www.microsoft.com/windowsphone/en-us/default.aspx>.
- [40] Mysql vs. postgresql benchmarks. In *Projecto Random Bugs*. Junho 2009. <http://www.randombugs.com/linux/mysql-postgresql-benchmarks.html>.
- [41] Brian Leroux Andre Charland. Mobile application development: Web vs. native.
- [42] Chyi-Ren Dow Cheng-Min Lin, Jyh-Horng Lin and Chang-Ming Wen. Benchmark dalvik and native code for android system. 54(5):49–53, 2011.
- [43] George Lawton. Lamp lights enterprise development efforts. 38:18–20, September 2005.
- [44] Toyotaro Suzumura Akihiko Tozawa Scott Trent, Michiaki Tatsubori and Tamiya Onodera. Performance comparison of php and jsp as server-side scripting languages. *IBM Tokyo Research Laboratory*.
- [45] Yichang China Cai Yi Xiaosheng Yu Coll. of Comput. & Inf. Technol., China Three Gorges Univ. Design and implementation of the website based on php & mysql. In *E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference on*, page 4, 7-9 Nov 2010.