



Instituto Politécnico  
de Castelo Branco  
Escola Superior  
de Tecnologia

## **Estrada Segura - AAF**

### **Aplicação Android e Alertas no Facebook**

Luis Manuel Pinto da Costa

Licenciado em Engenharia Informática

#### **Orientadores**

Professor Doutor Nuno Filipe Alves Gaiola Castela

Dissertação apresentado à Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Desenvolvimento de Software e Sistemas Interativos, realizada sob a orientação científica do Professor Doutor Nuno Filipe Alves Gaiola Castela, do Instituto Politécnico de Castelo Branco.

**Abril 2015**







## Composição do júri

### Presidente do júri

Doutor Alexandre José Pereira Duro da Fonte,

Professor Adjunto da UTC de Informática da Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco.

### Vogais

Doutor Nuno Manuel Garcia dos Santos,

Professor Auxiliar da Universidade da Beira Interior.

Doutora Mónica Isabel Teixeira da Costa,

Professor Adjunto da UTC de Informática da Escola Superior de Tecnologia de Castelo Branco, do Instituto Politécnico de Castelo Branco.

Doutor Nuno Filipe Alves Gaiola Castela,

Professor Adjunto da UTC de Informática da Escola Superior de Tecnologia de Castelo Branco, do Instituto Politécnico de Castelo Branco (Orientador).



## **Dedicatória**

Filipa Peleja, família e amigos.





## Agradecimentos

Agradeço ao meu orientador, Nuno Castela, pela disponibilidade, orientação, ajuda, crítica e principalmente pelo encorajamento quando o tempo restante era escasso para concluir esta etapa.

Agradeço a todos os meus colegas que, de alguma forma, colaboraram e me ajudaram no desenvolvimento deste trabalho, em especial ao Bruno Ramos, Marco Ferreira e Bruno Madaleno.

Agradeço aos meus amigos por todo o apoio, ajuda, sugestões e testes à aplicação. Um agradecimento especial ao Élio Cariano pela ajuda na gestão dos grupos do Facebook.

Um agradecimento especial para os meus pais e irmã por todo o carinho, ânimo, incentivo e por estarem sempre presentes.

O agradecimento mais importante é para a minha namorada, Filipa Peleja, que sempre me apoiou e ajudou ao longo desta caminhada. É graças a ela que hoje estou a atravessar a meta e a concluir com sucesso esta etapa.

A todos um Muito Obrigado!

Luis Costa



## **Resumo**

A popularidade dos dispositivos móveis tem vindo a crescer a um ritmo muito elevado. Em particular a possibilidade de localizar o dispositivo móvel trouxe uma mais-valia para uma variedade muito grande de diferentes aplicações. O valor de poder obter a localização marcou o futuro da nova geração de telemóveis. Serviços de localização podem ser utilizados para melhorar a informação de tráfego, onde poderá superar a informação obtida por sistemas tradicionais. O objetivo desta dissertação é disponibilizar um trabalho de investigação que examina uma solução em como detetar perigos na via pública.

O primeiro problema que é investigado é a identificação da posição do dispositivo móvel. É feito um estudo sobre como calcular a posição do dispositivo em relação aos perigos próximos. O segundo problema abordado é o da apresentação da interface ao utilizador. A experiência do utilizador com a aplicação é devesas importante e poderá determinar o sucesso, ou insucesso, de uma aplicação. Além de ser uma aplicação que precisa de informar o utilizador de um perigo na proximidade sem causar distração. Finalmente, foi feito um estudo sobre a integração de uma rede social com a aplicação de forma a atingir um maior impacto sobre a população. Os resultados da utilização da aplicação mostraram um maior pico quando foi realizada a integração com a rede social Facebook.

## **Palavras chave**

Android, smartphone, padrões de sincronização, segurança.



## **Abstract**

Mobile phones popularity has been increasing at a very fast rate. In particular, the possibility of tracking a mobile device has surged as a valuable asset for a large number of applications. The diversity of advantages from using tracking information as set a milestone on the future generation of mobile phones. Tracking services can be used to improve traffic information in which they may overpower traditional systems. The objective of the present thesis is to research a solution on how to detect dangers on public areas, more specifically roads.

The first problem investigated is how to use the information of the position of the mobile phone. The investigation starts with a comprehensive study in how to measure the most correct distance between the mobile device and the nearby dangers. A second problem is how to present the interface to the user. The user experience with the application is highly important and may be determinant for the success, or failure, of the application. Besides it is important to keep in mind that it is an application that must inform about nearby danger without distracting the driver. Moreover, it was performed a study on the impact of integrating a social network module in the application. It has been observed a higher usage as a result of adding this module.

## **Keywords**

Android, smartphone, synchronization patterns, safety.



# Índice geral

Introdução.....	1
1.1. Contexto e motivação .....	1
1.2. Objetivo.....	2
1.3. Formalização do problema.....	3
1.4. Organização do documento.....	5
2. Estado de Arte.....	7
2.1. Padrões para sincronização de dados.....	8
2.2. Armazenamento de dados e padrões disponíveis.....	11
2.3. Tráfego em dispositivos móveis.....	14
3. Metodologia da Investigação.....	17
3.1. Questões éticas.....	18
4. Arquitetura.....	21
4.1. Arquitetura Cliente-Servidor.....	21
4.2. Arquitetura de 3-Tier .....	22
4.3. Esquema geral.....	23
4.4. Tecnologias usados na aplicação .....	25
5. Modelação da Aplicação .....	29
5.1. Processo de modelação.....	29
5.2. Análise de requisitos.....	32
5.2.1. Requisitos funcionais.....	32
5.2.2. Funcionalidade de um Sistema Estrada Segura em Portugal.....	32
5.2.3. Recolha de informação .....	32
5.2.4. Integração com redes sociais – Facebook.....	33
5.3. Casos de uso .....	34
5.4. Diagrama entidade relacionamentos .....	40
5.5. Descrição de tabelas .....	43
6. Estrada Segura .....	49
6.1. Distância entre marcos .....	49

6.2.	Cálculo do distrito .....	52
6.3.	Estrada Segura na loja play (Google).....	56
7.	Usabilidade da Interface.....	59
8.	Conclusão .....	67
8.1.	Trabalho Futuro.....	68
	Referências Bibliográficas.....	69



## Índice de figuras

<b>Figura 1</b> - Sincronização de dados assíncrona.....	9
<b>Figura 2</b> - Sincronização de dados síncrona.....	11
<b>Figura 3</b> - Armazenamento parcial.....	12
<b>Figura 4</b> - Armazenamento total.....	14
<b>Figura 5</b> - Aplicação móvel para informação de tráfego.....	15
<b>Figura 6</b> - Processo de Investigação - Método Científico (hipótese-dedução) ..	18
<b>Figura 7</b> - Arquitetura Cliente-Servidor.....	21
<b>Figura 8</b> - Arquitetura de 3-Tier.....	23
<b>Figura 9</b> - Arquitetura do sistema em 3 camadas.....	23
<b>Figura 10</b> - Arquitetura do sistema com representação das camadas e subcamadas.....	24
<b>Figura 11</b> - Arquitetura da Aplicação Móvel.....	24
<b>Figura 12</b> - Modelo Castata ( <i>Waterfall model</i> ).....	30
<b>Figura 13</b> - Modelo Incremental.....	31
<b>Figura 14</b> - Casos de uso da aplicação “Estrada Segura”.....	35
<b>Figura 15</b> - Modelo ER servidor.....	41
<b>Figura 16</b> - Modelo ER integração com o Facebook.....	42
<b>Figura 17</b> - Modelo ER completo após integração com o Facebook.....	42
<b>Figura 18</b> - Modelo ER dispositivos móveis.....	43
<b>Figura 19</b> - Raio de sincronização de dados.....	50
<b>Figura 20</b> - Aplicação móvel com um acidente dentro do raio de alertas.....	51
<b>Figura 21</b> - Carta Administrativa Oficial de Portugal - CAOP.....	53
<b>Figura 22</b> - Botão para o grupo do Facebook correspondente ao distrito da minha localização.....	54
<b>Figura 23</b> - Evolução da aplicação desde Outubro 2013 a 1 Janeiro de 2014...	57
<b>Figura 24</b> - Integração da rede social Facebook na aplicação (3 Dezembro 2014). .....	57
<b>Figura 25</b> - Descarregamentos da aplicação Estrada Segura na loja play desde Outubro de 2013 a Fevereiro 2015.....	57

<b>Figura 26</b> - Interface da aplicação no estudo de usabilidade.....	66
<b>Figura 27</b> - Interface da aplicação após estudo de usabilidade.....	66

## Lista de tabelas

<b>Tabela I</b> - Caso de Uso – Adicionar Marco. ....	36
<b>Tabela II</b> - Caso de Uso – Confirmar Marco.....	37
<b>Tabela III</b> - Caso de Uso – Remover Marco.....	37
<b>Tabela IV</b> - Caso de Uso – Sincronizar Aplicação. ....	38
<b>Tabela V</b> - Caso de Uso – Editar Preferências.....	38
<b>Tabela VI</b> - Caso de Uso – Consultar Publicações Facebook.....	39
<b>Tabela VII</b> - Caso de Uso – Publica no Facebook.....	39
<b>Tabela VIII</b> - Dispositivo Móvel. ....	44
<b>Tabela IX</b> - MARKER_TYPE.....	45
<b>Tabela X</b> - MARKER_CONFIRMATION.....	45
<b>Tabela XI</b> - MARKER (Server). ....	46
<b>Tabela XII</b> - DISTRICT (Server). ....	46
<b>Tabela XIII</b> - CITY (Server).....	47
<b>Tabela XIV</b> - MARKER (Dispositivo móvel). ....	47
<b>Tabela XIV</b> - Número de utilizadores por distrito no Facebook.....	52
<b>Tabela XV</b> - Descrição dos participantes.....	61
<b>Tabela XVI</b> - Questionário sobre a aplicação Estradas Segura. As respostas são avaliadas numa escala de Likert de 1 a 5, onde 1 – Não Concordo e 5 – Concordo. A média corresponde à média aritmética e desvio padrão é uma medida estatística que identifica a variação em relação à média. ....	64
<b>Tabela XVII</b> - Interface da aplicação – novas funcionalidades. ....	65



## Introdução

### 1.1. Contexto e motivação

O problema de congestionamento de tráfego automóvel na via pública é um problema universal que apresenta um impacto muito elevado a nível pessoal, trabalho e segurança. Os atrasos e a inconveniência causada devido ao congestionamento das vias reduz a qualidade de vida das pessoas que têm de esperar, provocando perda de dinheiro para várias empresas e dificultando a rapidez de resposta dos pedidos de emergência. Portanto, se fosse possível atempadamente alertar a população que existe um problema na via e, e consequentemente, avisar que deverão utilizar caminhos alternativos terá um resultado muito positivo na vida de muitas pessoas. Uma forma de resolver este problema seria fornecer aos automobilistas informação em tempo real de forma a permitir uma resposta rápida e informada. Por vezes recebemos este tipo de informação através da rádio.

Recentemente a empresa Garmin<sup>1</sup> apresentou um produto com o seguinte *slogan* “*Don’t Hate Traffic – Avoid it*”. O produto utiliza tecnologias como DAB Radio, Smartphone Link e FM radio<sup>2</sup> para informar o automobilista da proximidade com uma via congestionada. Portanto resolver o problema de congestionamento de tráfego é um problema que chamou a atenção tanto da academia como da indústria. Relativamente a este produto da Garmin, foram identificadas duas limitações: para usufruir da tecnologia o utilizador terá de adquirir um dispositivo. Logo exige que seja feito um investimento monetário sobre algo que o utilizador ainda não tem conhecimento da sua mais-valia, e além disso, esta abordagem pressupõe que a via se encontre bastante congestionada para avisar o automobilista, pelo que não pode ser considerada uma abordagem preventiva. Quando existe um problema na via pública o automobilista poderá evitar a via ou ficar mais atento – o que não significa necessariamente que o trânsito já se encontre congestionado.

No entanto, nos dias de hoje observa-se um aumento de interação com interfaces tangíveis (Preece, Rogers, & Sharp, 2002), em particular, com dispositivos móveis. O número de pessoas que adquiriram um dispositivo móvel em comparação com o número de automobilistas que adquiriram um dispositivo como o da Garmin é muito mais elevado. E, por isso, apesar dos dispositivos da Garmin apresentarem esta funcionalidade, ainda não previnem um número de pessoas muito elevado. O potencial dos dispositivos móveis tem vindo a crescer ao longo dos anos, principalmente com o aparecimento dos *smartphones*. Este possibilitaram a

---

<sup>1</sup> <http://garmin.com/>.

<sup>2</sup> <http://www.garmin.com/en-GB/traffic/>.

implementação de uma variedade de diferentes aplicações e, conseqüentemente, a sua popularidade possibilitou o aparecimento de lojas que vendem aplicações apenas para este tipo de dispositivos (e.g. itunes da Apple) (Sherman, 2014). Por a presença dos dispositivos móveis ser constante existem muitos investigadores a estudar a ubiquidade <sup>3</sup> dos dispositivos móveis (Polatidis & Georgiadis, 2014; Zaharakis & Komninos, 2012).

## 1.2. Objetivo

No contexto da presente dissertação propõe-se desenvolver uma aplicação móvel que visa alertar o automobilista de possíveis perigos na via e, conseqüentemente aumentar o seu nível de alerta para que adotem comportamentos mais adequados no contexto da circulação rodoviária.

O trabalho proposto consiste no desenvolvimento de uma aplicação que tem como objetivo facilitar uma tarefa humana, mais especificamente, a condução de veículos motorizados, melhorando o seu conhecimento acerca do contexto rodoviário. A aplicação possui um modelo específico de interação que minimizará a aprendizagem da sua utilização. O desafio será conseguir promover a interação pessoa-máquina de forma natural e no futuro espera-se que esta interação se torne transparente – uma interação que seja baseada no conhecimento do seu contexto, e por isso, tem a capacidade alterar comportamentos. E, assim conseguirá capturar a atenção de muitos mais utilizadores (Abowd, 1999). Assim, será possível utilizar este sistema para influenciar os membros da sociedade a alterar comportamentos, com vista a tornar a circulação rodoviária mais segura.

Pretende-se promover um sistema onde se inserem localizações GPS onde existem: acidentes, veículos imobilizados, obstáculos na estrada, estradas bloqueadas, e outras situações de perigo. A inserção desta informação irá promover um comportamento mais atento ao meio envolvente e, conseqüentemente, um maior cuidado na circulação na estrada. Os automobilistas recebem um alerta se estiverem dentro de um raio de distância em relação à localização onde foi previamente inserido um alerta.

A aplicação não irá impor nenhuma autenticação. Esta decisão advém da observação da reação de vários utilizadores nas redes sociais. Pretende-se satisfazer dois grupos de pessoas que apresentam uma postura de utilização muito distinta: (1) um grupo que promove a sua privacidade, e (2) outro grupo que gosta de manter

---

<sup>3</sup> Refere-se a ser algo que está muito presente. Ou seja aparece em quase todo o lado.

uma reputação na comunidade e, por isso, é possível definir um *nickname* para a sua identificação. A importância da autenticação está implicitamente associada à interação da aplicação com a rede social Facebook<sup>4</sup>. O objetivo de integrar uma rede social com a aplicação proposta depreende-se pela necessidade de promover a aplicação, ou seja, agilizar a interação entre os utilizadores e aumentar o número de pontos de referência, ou alerta, inseridos na aplicação Estrada Segura. O desenvolvimento do trabalho proposto pressupõe o uso das seguintes tecnologias: *Android Google Maps API v2*<sup>5</sup>, *Spring Framework*<sup>6</sup>, *Spring Social*<sup>7</sup>, *PostgreSQL v9*, *SQLite* e *ORMLite*.

### 1.3. Formalização do problema

A motivação por detrás do trabalho proposto é clara, pois trata-se de uma necessidade que tem vindo a ser pensada desde há muito tempo. Contudo existe um conjunto de problemas que será necessário abordar antes de utilizar uma aplicação como Estrada Segura:

- Problemas técnicos: a cobertura da rede é determinante para o bom funcionamento da aplicação.
- Devido à comunicação com o servidor a aplicação apenas será elegível para utilizadores que tenham um pacote de dados ativo.
- A questão da privacidade e segurança: utilizadores que não queiram uma aplicação a analisar os seus percursos diários.
- Inclusão de informação de outras fontes (e.g. Estradas de Portugal) poderá implicar custos adicionais.

Também, no desenvolvimento desta dissertação foram abordados várias questões:

- Desenho da aplicação: Usabilidade e rapidez de comunicação.
- O cálculo da proximidade a um marco. E cálculo do distrito a que o marco pertence.
- Garantir a fluidez de comunicação de dados: evitar colisões.

---

<sup>4</sup> Facebook. Disponível em: <http://www.facebook.com/>.

<sup>5</sup> Google Maps Android API v2. Disponível em: <https://developers.google.com/maps/documentation/android/>.

<sup>6</sup> Spring Framework. Disponível em: <http://projects.spring.io/spring-framework/>.

<sup>7</sup> Spring Social. Disponível em: <http://projects.spring.io/spring-social/>.

- Evitar envios de pacotes muito grandes devido ao consumo de dados do pacote do utilizador.
- Evitar realizar um número de comunicações muito elevado com o dispositivo. Para reduzir o impacto negativo no tempo de vida da bateria do telemóvel.
- Integração com redes sociais.

O problema que será abordado nesta dissertação pode ser descrito em três questões: a primeira questão tem como maior preocupação garantir a comunicação de dados (dispositivos móveis ↔ servidor) e, em tempo-real, calcular a proximidade do automobilista ao marco; a segunda questão tem maior foco no desenho da aplicação; e na terceira questão foi feita uma análise do impacto da inclusão de uma rede social na aplicação.

**Questão 1:** *“maior preocupação para garantir a comunicação de dados (dispositivos móveis ↔ servidor) e calcular em tempo real a proximidade de automobilista”*

- Objetivo: estudar técnicas que calculam proximidade entre pontos e encontrar uma solução.
- Objetivo: implementar algoritmos que tenham em atenção que poderá existir várias comunicações para o mesmo marco; receber várias comunicações em simultâneo;
- Objetivo: não sobrecarregar o servidor com pedidos se houver a possibilidade de utilizar a capacidade de processamento do dispositivo móvel. E tem a vantagem do utilizador utilizar menos pacotes de dados.

**Questão 2:** *“maior foco no desenho da aplicação e o impacto que tem seu sucesso”*

- Objetivo: minimizar a complexidade da aplicação para evitar distrair o automobilista.
- Objetivo: determinar qual será a melhor posição e dimensão dos botões para serem facilmente utilizados. Portanto exigirem pouca atenção para analisar a aplicação.
- Objetivo: apresentar os marcos e informação associada de forma clara.

**Questão 3:** *“impacto da inclusão de uma rede social na aplicação”*

- Objetivo: garantir a privacidade do utilizador ao integrar uma rede social na aplicação.
- Objetivo: divulgar os marcos e garantir uma maior adesão por parte da população.



## 1.4. Organização do documento

Para além do presente capítulo, a dissertação de mestrado encontra-se dividida da seguinte forma:

- No Capítulo 2 faz-se um enquadramento tecnológico, apresentam-se alguns dos problemas existentes no desenvolvimento de aplicações móveis. São ainda descritos alguns dos *patterns* existentes para a resolução destes problemas.
- No Capítulo 3 são apresentadas as metodologias utilizadas durante todo o processo. Neste capítulo abordam-se ainda questões éticas que foram tidas em conta.
- No Capítulo 4 é abordada a arquitetura da aplicação e as tecnologias utilizadas para no desenvolvimento da solução.
- No Capítulo 5 analisa-se o problema e apresenta-se a modelação da solução. É neste capítulo que são descritos os casos de uso e o modelo relacional da solução.
- No Capítulo 6 é descrita a aplicação, Estrada Segura, bem como algumas das características que a diferenciam.
- No Capítulo 7 são apresentados os testes feitos à usabilidade da aplicação e as melhorias feitas de forma a aumentar a usabilidade da mesma.
- No Capítulo 8 apresentam-se as conclusões, tiradas ao longo de todo o processo de desenvolvimento, bem como o trabalho futuro.



## 2. Estado de Arte

Na última década observamos um aumento do uso de dispositivos que acedem à Internet (Meeker, Pitz, Fitzgerald, & Ji, 2005). Os dispositivos podem ser *smartphones* ou *tablets*, e por isso, a necessidade de aplicações móveis tem vindo a ganhar relevância (Wang & Ma, 2014). Tecnologias como iOS (Apple), Android (Google), Windows Phone (Microsoft) surgiram neste mercado emergente e continuam a desenvolver plataformas que visam ser flexíveis com arquiteturas e plataformas *open-ended* (Ribeiro & da Silva, 2012). Com este objetivo, Apple<sup>8</sup> e Google<sup>9</sup> disponibilizaram tutoriais com informação sobre as melhores práticas para produzir aplicações para estes dispositivos contudo ainda não existe consenso numa coleção de *design-patterns* ou *pattern language* para desenvolvimento de aplicações móveis (Ohrt & Turau, 2012). Devido a isto surgem dificuldades em produzir soluções neste domínio. Nesta secção irei discutir alguns destes aspetos, a solução desenvolvida será descrita em maior detalhe no capítulo 5.

Muitas aplicações móveis têm como objetivo transformar-se de forma portátil em mapas, dicionários, bibliografias digitais, etc. Por se tratar de uma tecnologia muito recente, as alterações são constantes por surgirem tecnologias que previamente não estarem disponíveis para este tipo de dispositivos. Uma questão muito importante está relacionada com os padrões para sincronização dos dados. É uma questão que exige muita atenção pois tem-se de garantir a consistência dos dados desde o dispositivo móvel ao servidor que armazena a informação (e o inverso também) (Franzago, Muccini, & Malavolta, 2014).

Google Maps é um exemplo de uma aplicação onde é impossível armazenar todos os dados no dispositivo (Li & Zhijian, 2010). Por isso é essencial que sejam implementadas mecanismos de sincronização para se obter os dados necessários. Existem outras aplicações que permitem que os utilizadores atualizem o seu perfil, ou que definam estratégias – aplicações relacionadas com a bolsa de valores – e por isso é deveras importante que as mesma se encontrem atualizadas com os dados mais recentes. Estes exemplos apontam para a necessidade de haver uma sincronização de dados como uma coleção de padrões, agrupados de acordo com os problemas que têm como objetivo resolver.

---

<sup>8</sup> <http://developer.apple.com/library/ios/#documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html>.

<sup>9</sup> <http://developer.android.com/design/index.html>.

## 2.1. Padrões para sincronização de dados

Padrões para sincronização de dados tem como objetivo ajudar na tarefa de decidir o momento adequado para sincronizar os dados entre um dispositivo móvel e o servidor (sistema remoto). Esta questão representa um problema comum que por vezes não é dada a devida importância aquando do desenho da aplicação móvel. Portanto, programadores de aplicações móveis devem ter em consideração um conjunto de limitações: disponibilidade da rede móvel, requisitos de atualização de dados, e a interface da aplicação (UI – *user interface*) (Alencar & Cowan, 2011).

Existem dois mecanismos para a sincronização de dados: *uploading* e *downloading*. *Uploading* refere-se à transferência de dados da aplicação móvel para um servidor; e, *downloading* à transferência de dados da aplicação móvel para o servidor. Em ambos os casos, se a comunicação falhar o utilizador deverá ter essa informação diretamente, ou indiretamente. Ou seja, através de uma notificação ou um registo que seja processado em separado pela aplicação. Os mecanismos de sincronização de dados são frequentemente padrões de arquiteturas. Um padrão de uma arquitetura indica uma forma fundamental de desenhar um diagrama da estrutura de uma organização para um sistema de *software*. Para tal, utiliza um conjunto de sistemas predefinidos onde é especificado responsabilidades, regras e indicações para organizar as relações entre os mesmos (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996).

O desafio é conseguir utilizar aplicações móveis e ter um acesso rápido aos dados (Giguère, 2001; Hyun & Kim, 1995). A capacidade de resposta e tempo de espera é essencial para determinar o quão rápido os dados podem ser acedidos num ambiente móvel. Uma aplicação que não seja responsiva ou muito lenta a dar *feedback* é indicativo de uma *user experience* (UX) de pouca qualidade. No caso de a aplicação ter um bom tempo de resposta na obtenção dos dados mas apresentar um tempo de espera longo a processar essa informação, o utilizador irá apresentar bastante desagrado ao utilizar a aplicação. E por isso, a importância de assegurar que a aplicação não fica bloqueada quando a sincronização dos dados ocorre – este passo deverá ser transparente ao utilizador.

Tendo em consideração a necessidade de assegurar uma aplicação com uma boa usabilidade e capacidade de resposta (Nilsson, 2009). É necessário obter uma sincronização de dados assíncrona numa perspetiva de (1) *uploading* e (2) *downloading*: (1) no *workflow* da aplicação o próximo estado da interface ou funcionalidade da aplicação não pode depender do resultado dos dados *uploaded*. Por exemplo, no caso de uma aplicação que gere dados de serviços de redes sociais, a atualização do *status* pode ser iniciado para um conjunto de diferentes serviços. Estas operações de *uploading*, que ocorrem simultaneamente, devem permitir que o utilizador possa continuar a interagir com a aplicação, ou até mesmo outras

aplicações, se o dispositivo móvel suportar *multi-tasking*, enquanto transfere a informação para atualizar o *status*. Logo a aplicação não depende do resultado das transferências a decorrer; (2) apesar de ser preferível utilizar dados que foram atualizados recentemente, uma aplicação pode utilizar um fração de dados estáveis para funcionar. Esta questão depende da natureza dos dados. Por exemplo, uma aplicação que gere a informação de uma conferência, e por isso, sendo transparente ao utilizador terá de atualizar a informação das horas e localização com um servidor e sendo esta informação mais recente. No entanto, dados como sumários das apresentações a realizar na conferência e informação mais estática não necessita de ser atualizada com a mesma frequência. E, se considerarmos uma aplicação que indica menus de restaurantes, que variam com muita pouca frequência, não fará sentido piorar a usabilidade do utilizador ao bloquear a interface com uma sincronização síncrona (ao contrário de assíncrona, que neste trabalho defendo ser a melhor opção) cada vez que a aplicação é aberta. Portanto, o ideal será realizar uma atualização (*download*) assíncrono paralelo à interface do utilizador.

Para alguns casos onde não existe muita memória disponível a sincronização assíncrona poderá não ser a melhor solução. Contudo esta era uma limitação mas evidente para versões de telemóveis mais antigas (e.g. primeiras versões do *Symbian* (Aapo, 2008)).

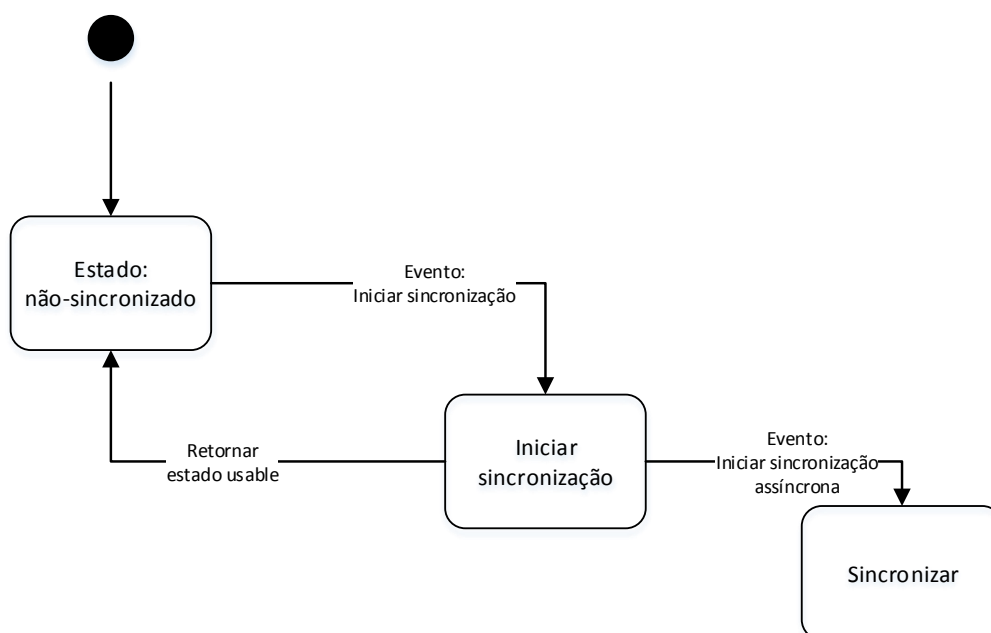


Figura 1 - Sincronização de dados assíncrona.

A sincronização de dados assíncrona é um mecanismo que quando a aplicação se encontra no estado *usable* (Figura 1) o utilizador pode interagir com a aplicação e o evento de sincronização pode iniciar. No entanto, invés de iniciar o evento de

sincronização sequencialmente o evento é iniciado assincronamente, logo retorna de imitado ao estado *usable*. Esta ação é possível via *trigger*, no sistema Android é usado o “intent”<sup>10</sup> ou no sistema iOS *push notification*. Para o mecanismo de notificação o Android e iOS utilizam “toasts”<sup>11</sup> e “UIAlertViews”<sup>12</sup> respectivamente. Finalmente, para informar que o sistema já acabou a transferência de dados Android e iOS utilizam o “intent” ou uma função de retorno (Murphy, 2010; Wenderlich et al., 2013). O trabalho proposto nesta dissertação utilizou um sistema assíncrono para Android. No entanto devido à importância e impacto do Android e iOS tenho como objetivo implementar a aplicação para iOS (Goadrich & Rogers, 2011).

A opção pela comunicação assíncrona deveu-se a dois motivos principais: A aplicação continuar disponível aquando da sincronização. Intuitivamente, a data mais recente não estará disponível, mas para muitas das situações o utilizador poderá interagir com a aplicação enquanto os dados sincronizam. E no caso dos dados já se encontrarem atualizados a UX do utilizador não será degradada devido de se encontrar à espera da atualização. E, sincronização de dados de forma transparente (*background*). Se o sistema *triggers* um evento de sincronização (via Android “intent” ou iOS “push notification”) enquanto a aplicação não está em primeiro plano, a aplicação pode sincronizar e atualizar os dados. Assim quando o utilizador retornar à aplicação esta estará atualizada. Contudo existem questões importantes a ter e atenção ao utilizar sincronização assíncrona:

1. Inconsistência de dados por acessos concorrentes à base de dados.
2. Muitos dados a serem carregados poderão implicar custos elevados ao utilizador.
3. Quantidade de dados a comunicar na rede poderão congestionar a rede da operadora móvel.

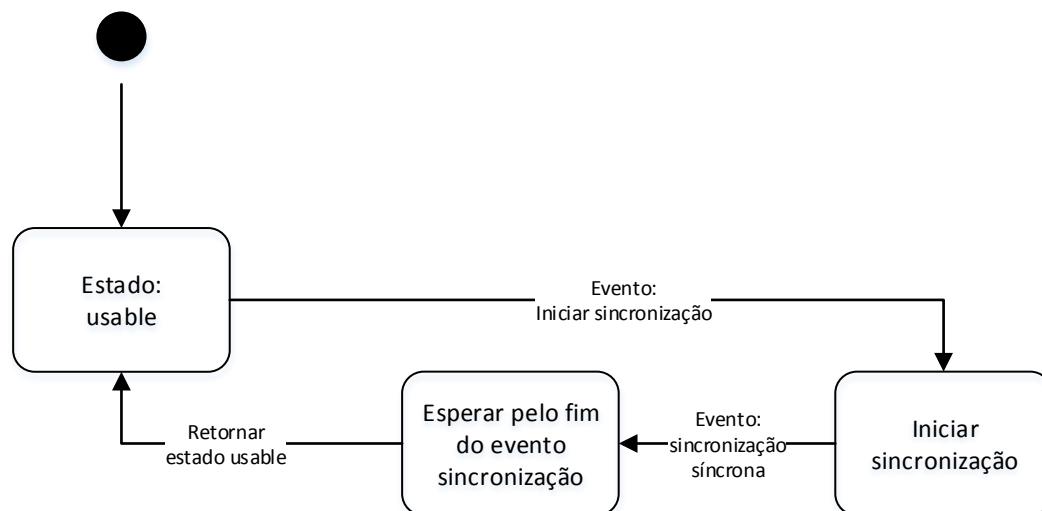
A sincronização de dados síncrona faz a gestão dos dados sincronizados, ou seja, cada vez que sincroniza os dados a interface do utilizador fica bloqueada. A sincronização síncrona é adequada quando existem aplicações que são dependentes de base de dados que têm de estar atualizados (*real time*) e precisos. Algumas aplicações móveis podem depender na resposta de uma ação antes de determinar qual será a ação seguinte. Portanto, para evitar que a aplicação avance para um estado que não é o correto, ao desenvolver a aplicação terá de se garantir que a aplicação ficará bloqueada até a sincronização de dados ter terminado.

---

<sup>10</sup> <http://developer.android.com/reference/android/content/Intent.html>.

<sup>11</sup> <http://developer.android.com/guide/topics/ui/notifiers/toasts.html>.

<sup>12</sup> [http://developer.apple.com/library/ios/#DOCUMENTATION/UIKit/Reference/UIAlertView\\_Class/UIAlertView/UIAlertView.html](http://developer.apple.com/library/ios/#DOCUMENTATION/UIKit/Reference/UIAlertView_Class/UIAlertView/UIAlertView.html).



**Figura 2** - Sincronização de dados síncrona.

O padrão de sincronização de dados síncrona, como a assíncrona, é um mecanismo de sincronização de dados. Quando uma aplicação está no estado *usable* o evento de sincronização pode iniciar (Figura 2). O evento é iniciado e executado sequencialmente, ficando a aplicação em espera que a sincronização do evento termine. A aplicação apenas retoma ao estado *usable* quando o evento sincronização termina.

A vantagem de utilizar a sincronização síncrona é que a aplicação poderá ser gerida muito mais facilmente do que com a comunicação assíncrona. Uma das vantagens, em comparação com a sincronização assíncrona, é que reduz o número de estados que a aplicação poderá estar num dado momento. A desvantagem, como descrita anteriormente, é que o tempo de espera poderá reduzir substancialmente a qualidade da experiência do utilizador (UX) aquando da utilização da aplicação. No entanto, existem aplicações como *Spotify* que é bastante popular que utiliza este tipo de sincronização. Neste caso, a aplicação apenas atualiza os dados do utilizador de acordo com o pagamento da respetiva conta. Em comunicação síncrona valida se o utilizador tem a conta paga e, apenas após confirmar esta informação faz atualiza os dados ou alerta que o utilizador expirou o tempo de validade do pagamento efetuado.

## 2.2. Armazenamento de dados e padrões disponíveis.

O armazenamento de dados e os respetivos padrões têm como objetivo ajudar a resolver os problemas de determinar qual serão os dados que devem ser guardados e os que deverão ficar armazenados sem haver fluxo de transferência de dados. Estas questões são importantes quando se desenha a aplicação móvel e os servidores que irá interagir. Usualmente as aplicações móveis têm limitações na

velocidade de internet, largura de banda disponível e capacidade de armazenamento (Kim, Ryu, & Ramachandran, 2012).

Uma solução é armazenamento parcial (McCormick & Schmidt, 2012). Para tal, os dados são sincronizados e armazenados apenas com os dados necessários para otimizar a largura de banda e o armazenamento de dados utilizado. A largura de banda e armazenamento são vitais para o desenho da aplicação sendo que muitas aplicações necessitam de ser adaptadas pois originalmente foram desenhadas para dispositivos que não são móveis (computadores, servidores, etc.) e onde os recursos são muito maiores. Nestas aplicações a otimização da solução era muitas vezes obtida por aumentar o uso da largura de banda e aumento da capacidade de armazenamento em disco, esta abordagem é impraticável para aplicações móveis. Um exemplo de uma aplicação é lojas de venda de livros, onde seria impraticável a aplicação ter memória todo o inventário de livros disponíveis. Portanto é necessário sincronizar a base de dados com os dados à medida que forem sendo precisos. Assim para a aplicação funcionar não é necessário ter armazenado a base de dados com toda a informação – uma informação parcial dos dados pode ser transferida apenas quando é necessária.

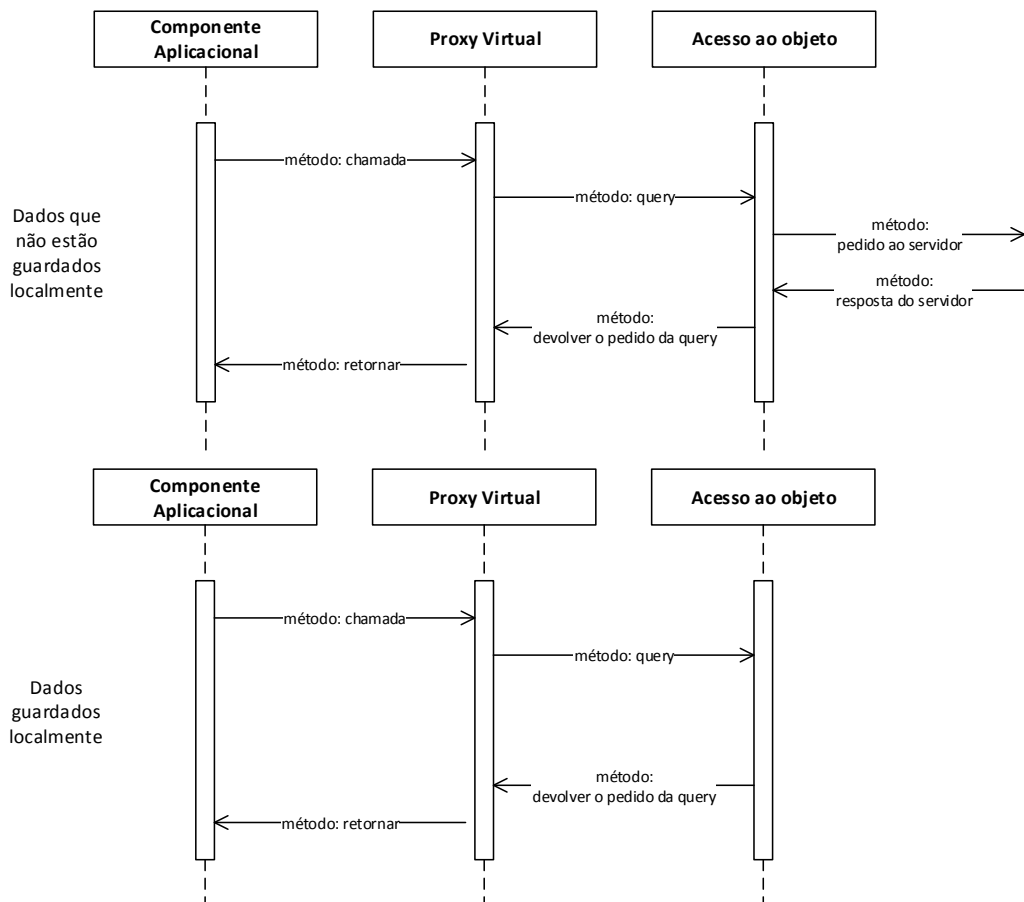


Figura 3 - Armazenamento parcial.



Na Figura 3 é apresentado o diagrama da sincronização para armazenamento parcial. Existem duas sequências que mostram como é processado para ambos os casos (dados guardados localmente e no servidor) os pedidos de mais informação. A primeira sequência refere-se ao caso de os dados não estarem guardados localmente e a segunda aos dados guardados localmente. Os dados são sincronizados dinamicamente *on-demand* por triggers da aplicação móvel, tipicamente utilizando o padrão de *Virtual Proxy* (proxy virtual) (Gamma, Helm, Johnson, & Vlissides, 1995). Um proxy virtual é um objecto com a mesma interface que o objecto usado pelo sistema que recebe as chamadas aos métodos, assim permite a inicialização dos campos apenas quando estes são necessários. Armazenamento parcial pode ser realizado utilizando os padrões do proxy virtual e o acesso ao objecto (Alur, Malks, & Crupi, 2001) que trata da sincronização a-pedido dos dados e não é apenas armazenamento local. O proxy virtual disponibiliza uma interface onde o objecto pode ser consultado pelas componentes e queries do acesso ao objecto (os dados irão preencher os campos do objecto). O acesso ao objecto é determina onde os dados estão disponíveis e apenas irá fazer pedidos ao servidor quando não estão disponíveis localmente. Este padrão permite minimizar os dados armazenados localmente mas também não necessita de utilizar a largura de banda para sincronizar com os dados todos. Assim, armazenamento parcial tem a vantagem de não utilizar demasiado espaço do dispositivo móvel e os dados podem ser sincronizados com diferentes níveis de granularidade: sincronizar apenas quando é necessário, é definido que dados e quantidade se pretende sincronizar. A desvantagem desta abordagem é que o utilizado para utilizar a aplicação terá de ter uma ligação à rede e estará dependente da largura de banda (velocidade) disponível.

Existem vários exemplos de aplicações que utilizam este tipo de abordagem. Há pouco foi mencionado uma aplicação de venda de livros mas também aplicações como Google Maps aplicam este tipo de padrões. O Google Maps apenas faz pedidos ao servidor para os mapas que são relevantes ao utilizador, e à medida que se faz zoom vão sendo feitos pedidos com um maior nível de granularidade. No caso da aplicação proposta também foi necessário aplicar este tipo de padrões e os motivos foi evitar: (1) sobrecarregar o dispositivo móvel com mapas que não são do interesse do utilizador; (2) procurar por alertas que não estão no raio de distância do utilizador; (3) gastar largura de banda e bateria do dispositivo móvel com atualizações que não estão no raio de distância do utilizador.

Em seguida será descrito o padrão armazenamento total. A análise deste padrão é indicativo do motivo porque não foi seleccionado para o trabalho proposto.

O armazenamento total sincroniza e guarda todos os dados necessários à aplicação – permite um tempo de resposta mais rápido a obter os dados e a carregar os dados na aplicação. Este padrão permite resolver problemas de quando a rede

não se encontra disponível ou não é desejável (e.g. *roaming*). Os dados (completos) são sincronizados com o dispositivo e o servidor num único evento. É uma abordagem ideal para aplicações que visam ter informação guardada no dispositivo no caso de não haver nenhuma rede disponível.

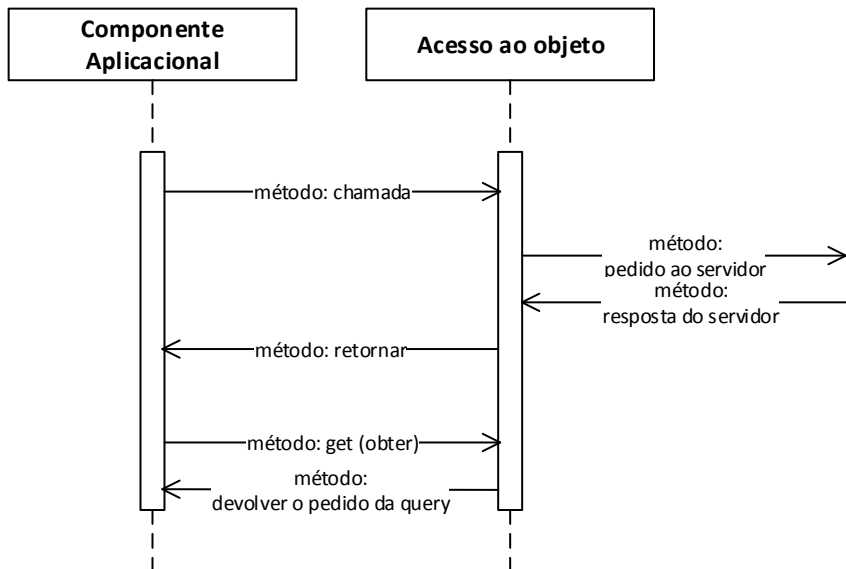


Figura 4 - Armazenamento total.

Na Figura 4 é apresentado o diagrama para o armazenamento total (McCormick & Schmidt, 2012). É feita uma distinta separação entre sincronização (método chamada) e o método que obtém os dados (*get*). A sincronização faz um pedido à rede e sincroniza com os dados (completos) que são posteriormente guardados no dispositivo. Todos os métodos *get* são feitos localmente no dispositivo móvel, ou seja, nunca é feita uma chamada ao servidor.

A vantagem do armazenamento total é o facto de se poder utilizar a aplicação mesmo quando não existe rede disponível. As desvantagens são: utilizar o armazenamento do dispositivo móvel, que pela natureza do dispositivo é limitado, e para aplicações que precisam de dados que são dependentes de novas atualizações é impraticável utilizar este tipo de armazenamento (e.g. aplicação Facebook).

### 2.3. Tráfego em dispositivos móveis

Na Figura 5 é apresentada uma proposta de uma aplicação para tráfego urbano. Este sistema utiliza a posição dos dispositivos móveis para colecionar dados em tempo-real, e obter uma estimativa do tráfego existente. Aplicações como MobiTraS (Manolopoulos, Tao, Rodriguez, Ismail, & Rusu, 2010) utilizaram este tipo de abordagem, em tempo-real: uma aplicação existente no dispositivo móvel processa os dados e envia a informação para um sistema central onde são executados os algoritmos (i.e. dados de localização, estimar o tráfego, etc.). A localização do

telemóvel pode ser calculado através das antenas GPS ou com a informação das antenas que fornecem sinal ao dispositivo. Naturalmente, este género de aplicações exigem que se tenha precaução em relação à privacidade do utilizador e, por isso, Manolopoulos et al. utilizaram a arquitetura GBA<sup>13</sup> por autenticação anónima (Manolopoulos, Papadimitratos, Tao, & Rusu, 2011). Como é possível verificar nesta abordagem toda o processamento de dados depende unicamente da posição do dispositivo. E por isso não contempla informação adicional fornecida pelo utilizador.

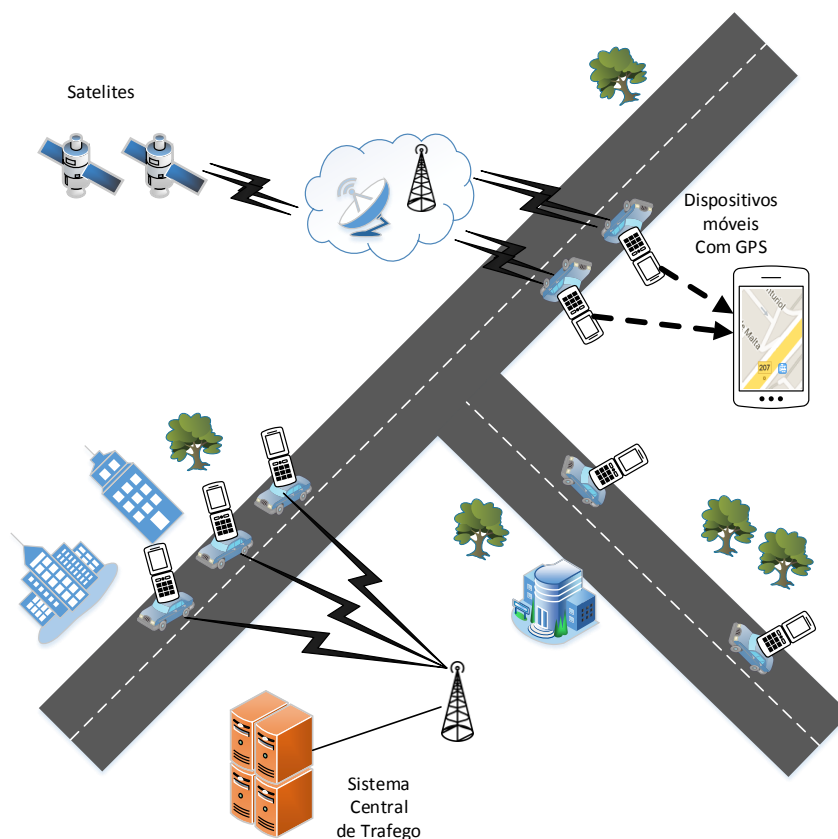


Figura 5 - Aplicação móvel para informação de tráfego.

<sup>13</sup> Do inglês, *Generic Bootstrapping Architecture*.

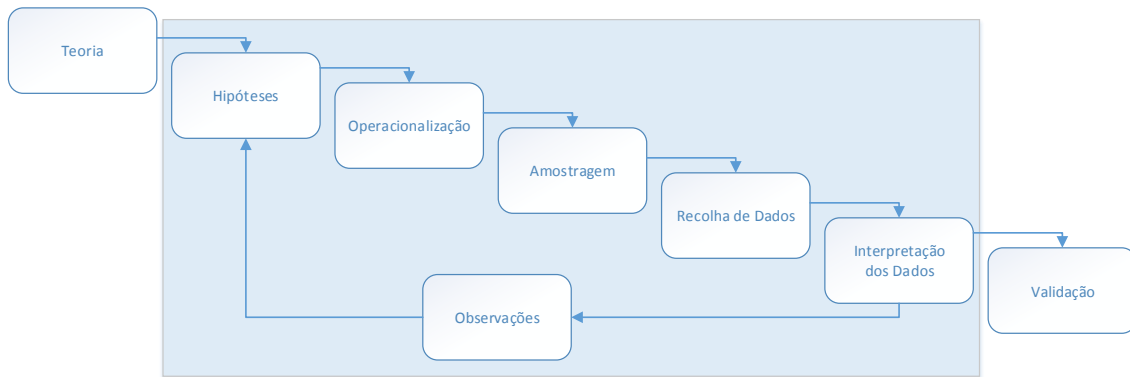


### 3. Metodologia da Investigação

O trabalho de investigação pode ser dividido em duas metodologias: qualitativa e quantitativa (Franklin, 2012; Venkatesh, Brown, & Bala, 2013). Se o objetivo for colecionar (i.e. *crawl* dados da Web) dados em larga escala será necessário elaborar teorias e hipóteses para se validar os mesmos. Usualmente os resultados da análise sobre os dados colecionados serão utilizados para criar novas hipóteses. Portanto existem estudos que o objetivo principal visa uma análise numa larga escala. No outro prisma, existe a metodologia qualitativa onde não existe uma quantidade de dados tão elevada mas existe uma maior preocupação a efetuar uma validação estatística sobre os dados que se está a trabalhar. É de ter em mente que estatística não responde a todas os problemas mas ajuda na compreensão dos dados. Os dados obtidos quantitativamente são usualmente obtidos de acordo com determinadas regras mas devido à sua quantidade a sua análise é menos subjetiva que os dados qualitativos, que muitas vezes são manualmente anotados.

No desenvolvimento do trabalho proposto foi realizada uma tarefa para avaliar a usabilidade da aplicação. Esta tarefa enquadra-se no método qualitativo – foram selecionadas 10 pessoas e avaliaram diferentes aspetos (eficácia, eficiência e satisfação) da aplicação. Porém, o trabalho apresenta uma maior proximidade ao método quantitativo. Isto porque o trabalho proposto pode ser observado com uma sequência de tarefas: (1) ideia a implementar; (2) elaboração das várias hipóteses em como desenvolver a ideia; (3) desenvolvimento e experimentação com alguns utilizadores; (4) e, de acordo com a observações do ponto (3) retoma-se ao ponto (2) até atingir um ponto onde se identifica que a aplicação estará pronta para validação. Portanto, para se realizar uma análise qualitativa do sistema Estrada segura, no ponto (4) o sistema teria de ser avaliado por pessoas que usufruíram de uma formação para saberem as boas práticas de como utilizar a aplicação. Ou seja para esta aplicação, um especialista será alguém que se encontra confortável com a tecnologia e que previamente a utilizar a aplicação teve uma formação – por exemplo, introdução sobre o que se pretende da aplicação e as melhores práticas de como a utilizar. Contudo a aplicação foi testada em larga escala por pessoas de diferentes idades, estratos sociais, formação, etc. E, por isso, não houve qualquer controlo sobre os indivíduos que descarregaram a aplicação. De acordo com (Flick, 2005) na investigação quantitativa: *“As situações em que os fenómenos e as relações estudadas ocorrem são controladas até ao limite do possível, a fim de determinar com máximo de clareza as relações casuais e a sua validade. Os estudos são desenhados de forma a excluir, na medida do possível, a influência do investigador”* e, o mesmo autor para a investigação qualitativa afirma que *“Ao contrário da investigação quantitativa, os métodos qualitativos encaram a interação do investigador como o campo e os seus membros como parte explícita da produção do saber, em lugar de a excluírem a todo o custo, como variável interveniente. A subjetividade do investigador*

*e dos sujeitos estudados faz parte do processo de investigação”* No entanto ao contrário de Flick existem autores (Kelle, 2005) que fundamentam que não deveria haver uma separação entre estas duas abordagens. (Cupchik, 2001) resume esta questão por considerar que as duas abordagens estão inter-relacionadas e, por isso, a abordagem quantitativa contribuí para a identificação de processos relevantes e, consequentemente disponibiliza uma investigação pela abordagem qualitativa.



**Figura 6 - Processo de Investigação - Método Científico (hipótese-dedução)**

A organização do processo de investigação pode ser esquematizado de acordo com Figura 6. O investigador elege uma teoria e para tal gera uma, ou mais, hipóteses que são elaboradas (operacionalização) e testadas (amostragem). Com base na interpretação dos dados obtidos o investigador tem duas opções: observa consistência nos resultados obtidos e avança para a validação; ou modifica as hipóteses e, como tal, retoma à etapa das hipóteses (Dodig-Crnkovic, 2002).

Na Figura 6 a etapa de validação corresponde ao processo onde uma medida de teste é usada para medir os resultados. Existem medidas quantitativas como: temporais, avaliações em exames, etc. Para o caso de uma aplicação com o Sistema Estrada Segura: número de utilizadores da aplicação ao longo do tempo; número de marcos diários; número de *likes* nos marcos; número de publicações por dia nos grupos do Sistema Estrada Segura na rede social Facebook (total e por distrito) e finalmente, número de pessoas nos grupos do Facebook.

### 3.1. Questões éticas

No processo de investigação surgiu a questão de ocultar, ou não, a informação do utilizador que inseriu o marco, logo uma questão de privacidade. Como os utilizadores se sentem em relação a isso?

Para um bom funcionamento da aplicação não é estritamente necessário que a inserção de um marco indique explicitamente aos outros automobilistas quem inseriu o marco. E, talvez alguns automobilistas prefiram o anonimato. No outro prisma existe a questão da reputação dos utilizadores da aplicação. Há

automobilistas que poderão contribuir mais porque gostam de ser gratificados com pontos validação dos seus marcos. E, ficando reconhecidos como automobilistas de confiança, respetivamente à inserção dos marcos. Neste caso são utilizadores que gostariam de ter um nome associado a si.

Devido a ser uma questão ambígua em que seria difícil de chegar a um consenso, optou-se por não ser feita uma validação exaustiva. O automobilista será associado a um *nickname* escolhido por si. Esse nome é escolhido pelo utilizador da aplicação. Sendo assim não se tem acesso a nenhuma informação sobre a pessoa inseriu o(s) marco(s).

Quando surge um novo marco – coordenada GPS no mapa – é inserido uma publicação automática no Facebook. A publicação apresenta um mapa com a indicação de onde ocorre o alerta e está associada ao comentário inserido pelo automobilista que criou o marco. No entanto, esta publicação não se encontra associada à conta da rede pessoal do Facebook do automobilista. As publicações automáticas são feitas pela aplicação Estrada Segura e, no respetivo Distrito que o marco se encontra associado.





## 4. Arquitetura

Arquitetura de computadores descreve a funcionalidade e a organização de como o sistema de computação está organizado. Portanto, de forma abstrata, define a capacidade de um computador e a sua organização “interna”. Arquitetura de computadores envolve vários aspetos, entre estes, desenhar a arquitetura, a lógica do desenho e a sua implementação. E, existem várias formas de desenhar uma arquitetura de rede de computadores. Refere-se a arquitetura de rede de computadores à forma como os computadores estão organizados num sistema e como as diferentes tarefas são alocadas aos mesmos. Duas arquiteturas mais populares são a arquitetura ponto-a-ponto (*peer-to-peer*) e cliente-servidor (*client-server*). A arquitetura cliente-servidor é também denominada por *tiered* porque utiliza várias camadas (Pyke & Blanc, 1973).

### 4.1. Arquitetura Cliente-Servidor

Nos primórdios da Web a arquitetura cliente-servidor foi a mais popular (Figura 7). O cliente – página web (*browser*) – efetua pedidos que são enviados e tratados pelo servidor (HTTP). O cliente tem de apenas receber as respostas e a apresentá-las ao utilizador. Portanto neste tipo de aplicações o servidor tem a responsabilidade do processamento de dados e, por isso, esta arquitetura é bastante simples. Atualmente é utilizada por muitos dos conteúdos existentes.

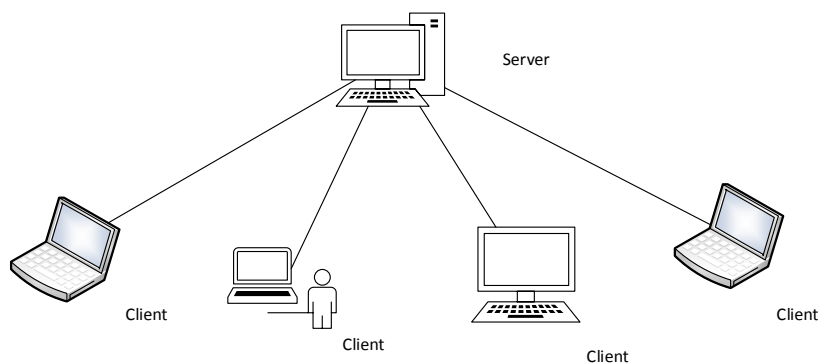


Figura 7 - Arquitetura Cliente-Servidor.

A vantagem da arquitetura cliente-servidor em relação a aplicações únicas (*stand-alone*) é a de se encontrar centralizada numa única máquina e, por isso, ser fácil de gerir. E, em simultâneo estar disponível por um número vasto de utilizadores – capacidade de comunicação pelo acesso à Web.

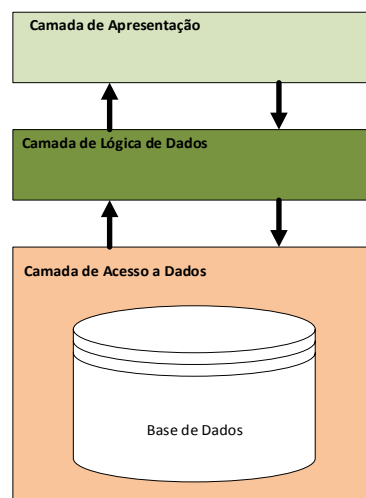
O problema do modelo cliente-servidor reside no facto de não existir uma distinção clara entre a lógica de processamento dos pedidos (conhecida como lógica de negócio) e o próprio sistema de gestão da informação. Em um cenário empresarial onde os sistemas de informação servem vários tipos de aplicações,

inclusivamente aplicações não-Web, a utilização de um sistema destes força a que cada aplicação tenha de redundantemente reimplementar a lógica de negócio. E assim, a gestão global do sistema poderá tornar-se numa tarefa extremamente complexa e, conseqüentemente, mais sensível à ocorrência de erros. Portanto, a arquitetura cliente-servidor constitui uma boa arquitetura para aplicações baseadas em páginas estáticas ou mesmo para aquelas que produzem informação dinâmica (no servidor) mas que não incorporam demasiado processamento para a sua execução.

## 4.2. Arquitetura de 3-Tier

A arquitetura de 3-Tier surge como forma de ultrapassar as limitações da arquitetura cliente-servidor. Nesta arquitetura, existem três zonas funcionais ou camadas distintas: a camada de apresentação suportada pelo cliente, a camada de lógica de negócio que reside no servidor e a camada de dados que pode encontrar-se ou não no mesmo servidor. A camada de apresentação tem exatamente as mesmas funções da arquitetura cliente-servidor, isto é, enviar pedidos e apresentar as respostas ao utilizador. No entanto, agora no lado do servidor existe uma divisão clara entre a lógica de negócio e a base de dados (Figura 8).

A função da camada intermédia (lógica de negócio) é a de proporcionar a construção dinâmica de informação que será enviada ao cliente. Esta análise é realizada sem preocupações de manutenção de integridade dos dados que agora são geridos independentemente. Esta separação permite que os mesmos dados possam ser facilmente acedidos por outras aplicações – que beneficiam das restrições de integridade que possam estar implementadas pela camada de dados. As aplicações terão de apresentar uma interface conveniente para esta comunicação. O mesmo se aplica à segunda camada, ou seja, agora é possível reaproveitar a lógica de negócio implementada desde que esta satisfaça os requisitos de outras aplicações que não precisam necessariamente de um interface Web.



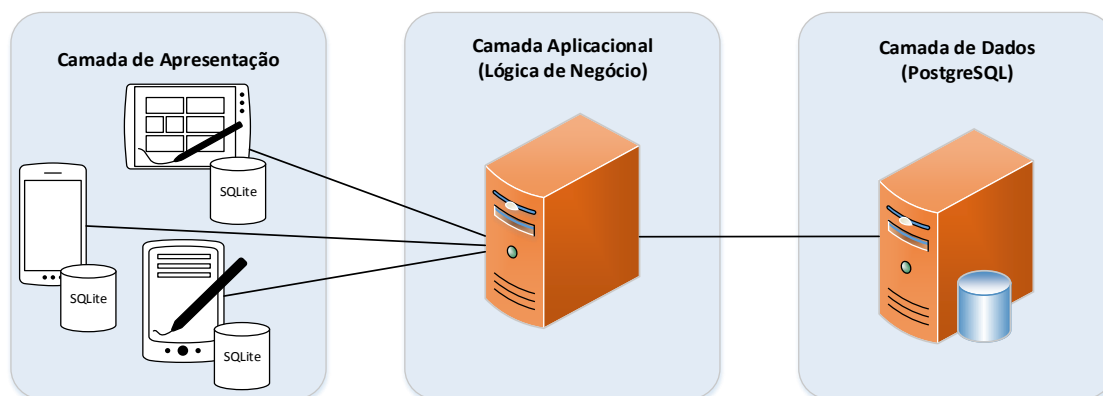
**Figura 8 - Arquitetura de 3-Tier.**

A arquitetura de 3-Tier indicia uma clara separação entre apresentação e conteúdo porém, na realidade, esta separação não é sempre assim tão clara. O facto de o HTML<sup>14</sup> ser o resultado do processamento intermédio efetuado pela camada intermédia implica que a camada lógica tenha de ter conhecimento de conceitos relativos à camada de apresentação – HTML é uma linguagem que também define a apresentação. Portanto, a solução passa por basear as respostas não na sua apresentação mas sim no seu conteúdo por utilização de uma linguagem como o XML. Desta forma, utilizando utilização de folhas de estilo diferentes, o mesmo tipo de informação poderia ser visualizado de forma mais apropriada ao cliente gráfico utilizado.

### 4.3. Esquema geral

A arquitetura proposta para o sistema da aplicação Estrada Segura está dividida em 3 camadas: camada de apresentação, a camada aplicacional e a camada de dados (Figura 9). Na Figura 10 é possível ver que a camada aplicacional se encontra subdividida em duas camadas, a camada de lógica de negócio e a camada de acesso a dados respetivamente.

O sistema esta dividido em dois módulos, aplicações, sendo eles a aplicação móvel e a aplicação servidor. A aplicação móvel esta dividida em 3 camadas, sendo elas a camada de apresentação, a camada de negócio e acesso a dados e a camada de dados (Figura 11).

**Figura 9 - Arquitetura do sistema em 3 camadas.**

<sup>14</sup> HTML é o acrónimo para HyperText Markup Language.

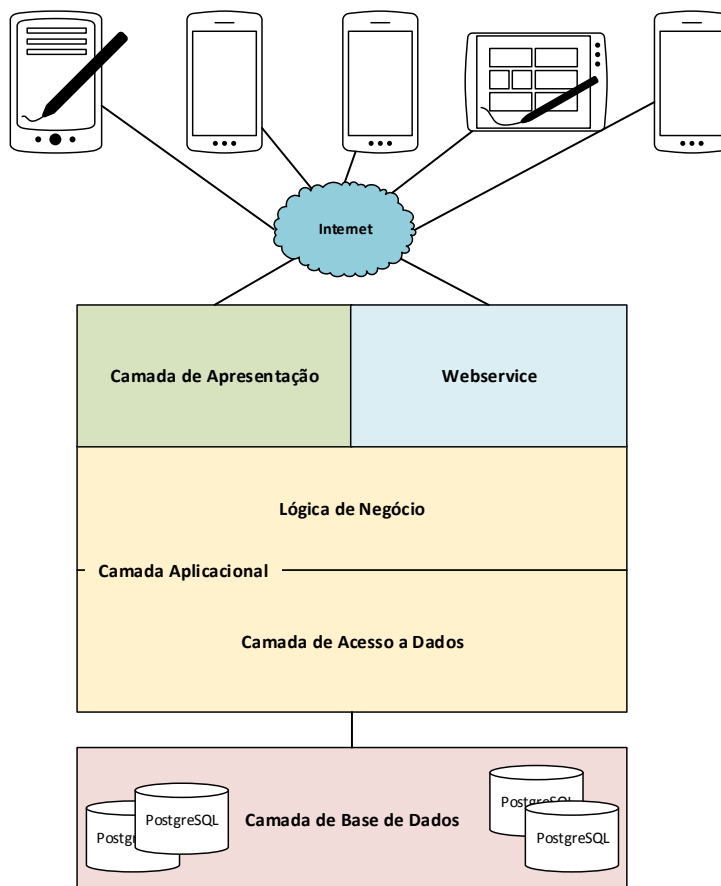


Figura 10 - Arquitetura do sistema com representação das camadas e subcamadas.

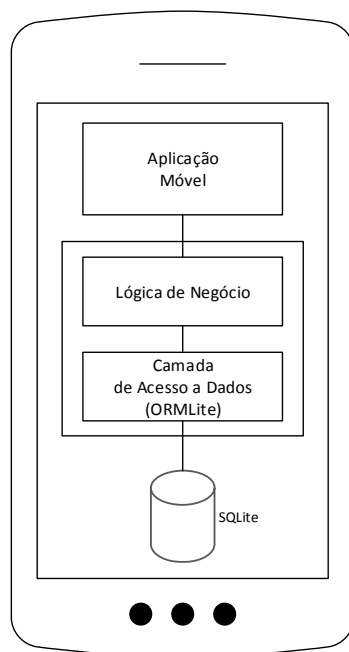


Figura 11 - Arquitetura da Aplicação Móvel.

#### 4.4. Tecnologias usados na aplicação

A aplicação móvel foi desenvolvida para ser utilizada na plataforma *Android*. A vertente do servidor foi desenvolvida para ser multiplataforma, isto é, poder ser instalada tanto em servidores com sistema operativo *Windows* ou *Unix*. No desenvolvimento de ambas as aplicações foi tido em consideração se as tecnologias eram ou não *open source*. De seguida são apresentadas as tecnologias utilizadas.

- **Android** é um sistema operativo (SO) móvel baseado em linux, desenvolvido pela empresa de tecnologia Google. Com uma interface com o utilizador baseada na manipulação direta, o Android é projetado principalmente para dispositivos móveis com ecrãs sensíveis ao toque como *smartphones* e *tablets*.

Android tem como objetivo ser utilizado num ecrã sensível ao toque, onde o utilizador pode manipular objetos virtuais (e.g. teclado virtual). Apesar de ser principalmente utilizado em dispositivos com tela sensível ao toque, também é utilizado em consolas de jogos, camaras, computadores e outros dispositivos eletrônicos.

- **SQLite** é uma biblioteca escrita na linguagem C que implementa uma base de dados SQL embebida. Programas que utilizam a biblioteca SQLite têm acesso a uma base de dados SQL sem executar um processo SGBD<sup>15</sup>. A biblioteca SQLite lê e escreve diretamente para a base de dados em disco.
- **ORMLite** é uma arquitetura *open source* que fornece um leve mapeamento objeto-relacional (ORM) entre as classes Java e uma base de dados SQL. Suporta base de dados JDBC<sup>16</sup>, bem como a plataforma móvel Android. ORMLite fornece uma camada que abstrai eficientemente as funções JDBC, evitando a complexidade de outras estruturas (e.g. Hibernate ou iBATIS). Esta arquitetura suporta ligações JDBC para MySQL, PostgreSQL, Microsoft SQL Server, H2, Derby, HSQLDB e SQLite. E também chamadas a bases de dados nativas do sistema operativo Android.
- **Google Maps Android API v2** – a Google disponibiliza na sua loja Google play a aplicação de mapas do Google (Google Maps) e também uma biblioteca (API) para programadores poderem utilizar a mesma. Google Maps disponibiliza imagens de satélite; percursos entre localizações: de carro, a pé e recorrendo a transportes

---

<sup>15</sup> Em português SGBD é o acrónimo de “Sistema de Gestão de Base de Dados”. O acrónimo em inglês é RDBMS “*Relational DataBase Management System*”.

<sup>16</sup> JDBC é o acrónimo para “*Java DataBase Connectivity*”.

públicos; perspetivas das ruas; etc. Uma das vantagens desta aplicação é a facilidade de poder ser integrada em outras aplicações ou páginas web. Neste momento é uma das aplicações mais populares para aplicações em *smartphones*. Num estudo realizado durante o mês de Agosto de 2013<sup>17</sup> 54% de todos os utilizadores de *smartphones* utilizaram pelo menos uma vez a aplicação de Google Maps.

- **JSON** é um formato leve para comunicação de dados. Este formato foi originalmente criado por Douglas Crockford e é descrito no RFC 4627<sup>18</sup>. O *media-type* oficial do JSON é "*application/json*" e a sua extensão é .json. Devido à sua simplicidade JSON é um muito formato utilizado tanto na academia como na indústria, apresentando-se como uma boa alternativa ao XML. Em comparação com o formato XML JSON apresenta-se como um formato mais leve, fácil de analisar e gerar.
- **Spring Data JPA** é uma implementação de uma camada de acesso a dados em uma aplicação. Usualmente o acesso a dados é moroso e complicado exigindo muito código para executar consultas simples (e.g. paginação e auditoria). Spring Data JPA melhora significativamente a camadas de acesso a dados e, conseqüentemente, reduz o esforço na fase de implementação.

As principais características da API Spring Data JPA são:

- Suporte para construir repositórios baseados em Spring e JPA.
  - Auditoria transparente das classes de domínio
  - Suporte para paginação, execução de consulta dinâmica, capacidade de integrar código personalizado para aceder a dados.
  - Mapeamento de entidades através de XML ou anotações.
- **Hibernate** é uma arquitetura ORM<sup>19</sup> escrita na linguagem Java. Também se encontra disponível para .Net com o nome NHibernate. Esta arquitetura facilita o mapeamento dos objetos de domínios (Java Entity) com as tabelas de uma base de dados. O mapeamento pode ser feito recorrendo a ficheiros (XML) ou anotações Java.

---

<sup>17</sup> GlobalWebIndex.

<http://www.businessinsider.com/google-smartphone-app-popularity-2013-9#infographic>.

<sup>18</sup> <https://www.ietf.org/rfc/rfc4627.txt>.

<sup>19</sup> ORM é o acrónimo para "*Object-relational mapping*".

- **Spring** é uma arquitetura *open source* para a plataforma Java, criada por Rod Johnson (Johnson, 2004). Spring é uma arquitetura baseada nos padrões de projeto de *inversion of control* (IoC) e injeção de dependência. O *container* de Spring é responsável por "instanciar" as classes de uma aplicação Java e definir as dependências entre elas.
- **Spring MVC** é uma arquitetura HTTP<sup>20</sup> baseada em *servlet*'s. Fornece um vasto numero de extensões para implementar aplicações web e/ou serviços web RESTful.
- **REST** é um acrónimo para *Representational State Transfer*. Trata-se de um estilo de arquitetura de *software* onde se aplicam as melhores práticas para gerar serviços web escaláveis. REST ganhou popularidade como sendo uma alternativa mais simples a SOAP<sup>21</sup> e WSDL<sup>22</sup>-based Web Services.
- **Tomcat** é um servidor web, mais especificamente um *servlet container*. Este servidor foi desenvolvido pela Apache Software Foundation<sup>23</sup> e distribuído como uma tecnologia *open source*. Tomcat implementa as tecnologias Java Servlet e JavaServer Pages (JSP) no entanto não se trata de um *container* EJB<sup>24</sup> (servidor aplicacional).
- **PostgreSQL** é um sistema que faz a gestão de base de dados relacionais. Originalmente desenhado para plataformas UNIX, contudo suporta ser utilizado em Mac os x, Solaris e Windows. É uma tecnologia *open source* e, por ser estável, necessita de pouco esforço de manutenção. Uma das vantagens é que PostgreSQL foi um dos primeiros programas que gerem base de dados, a implementar MVCC (Multi-Version Concurrency Control). Também permite a adição de funções feitas à medida em diferentes linguagens de programação: C/C++, Java, entre outras.

---

<sup>20</sup> Acrónimo para "*Hypertext Transfer Protocol*".

<sup>21</sup> Acrónimo para "*Simple Object Access protocol*".

<sup>22</sup> Acrónimo para "*Web Services Description Language*".

<sup>23</sup><https://www.apache.org/> .

<sup>24</sup> Acrónimo para "*Enterprise JavaBeans*".





## 5. Modelação da Aplicação

Os requisitos necessários são definidos durante a fase inicial da modelação do sistema. Estes requisitos descrevem como o sistema deverá funcionar, o seu domínio, limitações do sistema e especificações de atributos e patentes associadas ao sistema. De acordo com (Böckle, van der Linden, & Pohl, 2005) o processo de desenvolvimento de um sistema pode ser dividido em quatro atividades fundamentais: (1) especificar as funcionalidades e limitações (ou restrições) do sistema; (2) desenvolvimento do sistema tendo em atenção as funcionalidades e limitações identificadas em (1); (3) testar o sistema a fim de garantir que garante todos os pedidos inicialmente formulados; e (4) ter em atenção a evolução do sistema, ou seja, que o sistema seja adaptável às necessidades adicionais que poderão surgir. Neste capítulo será dada uma maior atenção a (1) – análise funcional do sistema.

O objetivo deste capítulo é identificar os requisitos necessários para implementar a aplicação Estrada Segura. Estes têm como objetivo identificar de forma clara como o dispositivo móvel comunica com o servidor e a rede social Facebook. É importante realçar a importância desta fase: é um dos esforços mais importantes no desenvolvimento de um sistema. Estes requisitos são desenvolvidos para garantirem que todo o esforço do desenvolvimento do sistema contempla o modelo conceptual que foi desenhado cuidadosamente para que todas as hipóteses de interação com o sistema decorressem sem haver falhas. Ou necessidades de refazer desenvolvimento por falta de análise.

### 5.1. Processo de modelação

No desenvolvimento de *software* não existe um processo de desenvolvimento de *software* mais ou menos correto. No processo de seleção do processo de desenvolvimento é tido em atenção o quão é adequado à complexidade do sistema, a equipa disponível para desenvolver e como será feita a utilização do sistema. Aquando da escolha do modelo de processo deve ser tido em consideração (Boehm & Belz, 1990):

- Desenvolvimento incremental – ao longo do desenvolvimento será feito um conjunto de incrementos de funcionalidades. Usualmente os incrementos surgem devido a necessidades que surjam posteriormente; restrições de orçamento sendo os incrementos desenvolvidos gradualmente; requisitos que tenham sido mal especificados e aplicações com um elevado número de módulos.
- Desenvolvimento orientado ao custo ou por tempos – é gerado uma lista de prioridades dos requisitos do sistema até atingir o limite de custo ou tempo disponível.

O modelo clássico é o modelo cascata (do inglês, *Waterfall model*) onde o desenvolvimento é linear e sequencial. O desenvolvimento em cascata segue as seguintes etapas: levantamento dos requisitos; desenho do sistema; testes (validação); instalação e integração do sistema; e, por fim, operação e manutenção (Figura 12).

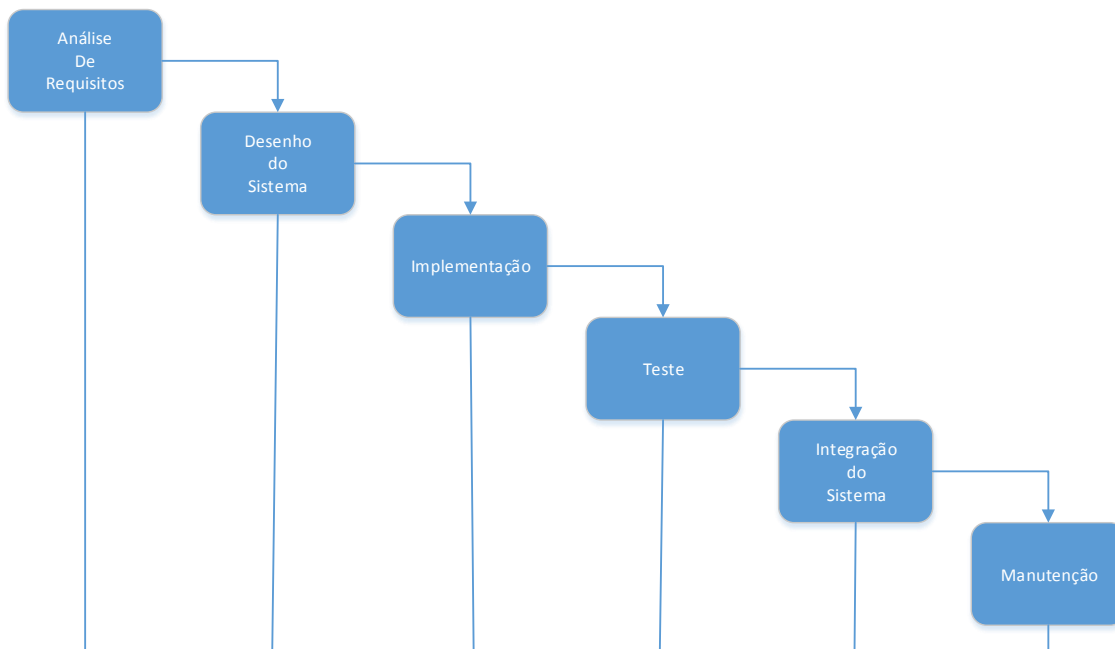


Figura 12 - Modelo Castata (*Waterfall model*)

A vantagem deste modelo é que existe uma separação clara entre cada fase de desenvolvimento; simples de compreender e utilizar; fácil de gerir (devido à rigidez do modelo); funciona bem para pequenos projetos com poucos requisitos; fases claramente definidas; e, os processos e resultados podem ser facilmente bem documentados. A desvantagem deste modelo é o de não permitir flexibilidade, por a sua natureza ser sequencial torna-se muito rígido quando é necessário retornar para uma das fases anteriores. Existem situações onde são detetados erros na modelação dos requisitos ou o cliente tem dificuldades em definir as suas necessidades explicitamente. Portanto, no desenvolvimento de *software* existe uma elevada quantidade de risco e incerteza; o modelo poderá ficar pobre para projetos que necessitem de incrementos; e muito complicado para realizar alterações nos requisitos.

Uma alternativa ao modelo sequencial é o modelo incremental, onde o modelo é desenhado, implementado e testado de forma incremental. Nestes modelos são adicionadas funcionalidades recursivamente até o sistema estar completo. Ao contrário do modelo sequencial onde a documentação se encontra muito bem

definida, no modelo incremental poderá ser complicado garantir esta funcionalidade. A vantagem de usar o modelo incremental é por este modelo permitir que seja dividido em vários “mini” projetos de desenvolvimento que poderão ser continuamente evoluídos ao logo do desenvolvimento do sistema e, por isso, permite que requisitos de maior prioridade sejam rapidamente processados. Em comparação ao modelo sequencial, o modelo incremental é mais flexível e reduz o risco de satisfazer os requisitos da aplicação.

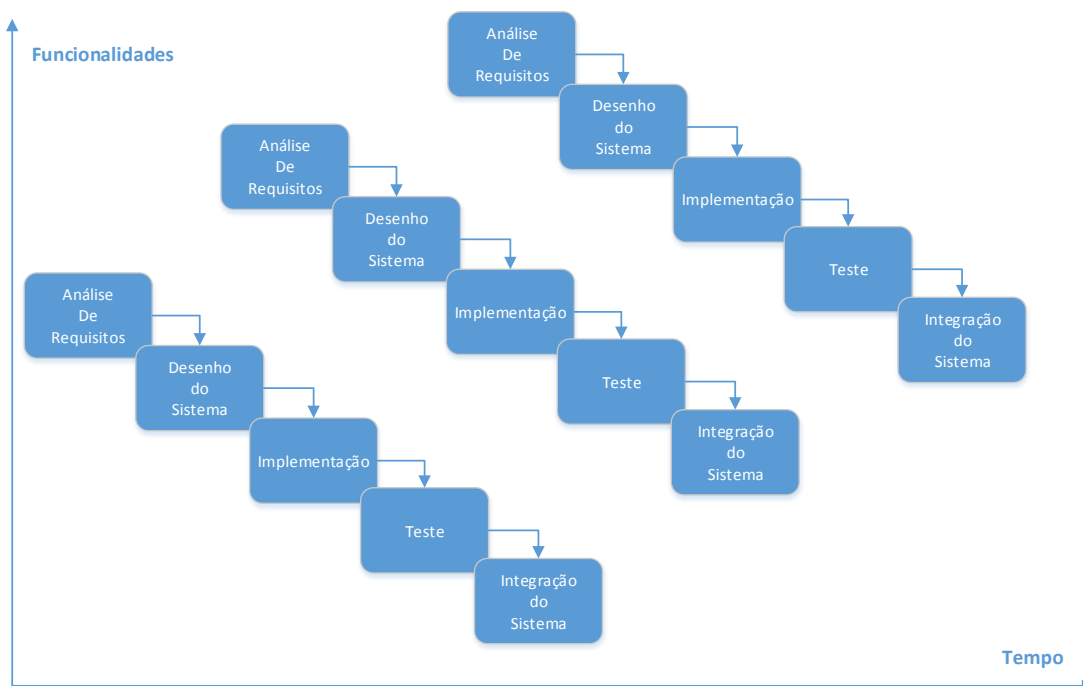


Figura 13 - Modelo Incremental.

No desenvolvimento do Sistema Estrada Segura foi utilizado o modelo incremental. O modelo em cascata seria uma boa opção pelas suas vantagens mencionadas anteriormente, no entanto, o Sistema Estrada Segura é uma aplicação de pequena dimensão e por isso facilita na gestão da documentação. E também não existem várias equipas a desenvolver tarefas que precisem de estar linearmente separadas. Contudo o motivo principal por optar pelo modelo incremental foi devido à necessidade de continuamente alterar os requisitos do sistema. As funcionalidades da aplicação foram sendo alteradas de acordo com necessidades que foram posteriormente identificadas. Como por exemplo, integração com a rede social Facebook, gosto/não gosto de um marco e adicionar as camaras de trafego.

## 5.2. Análise de requisitos

### 5.2.1. Requisitos funcionais

Em engenharia de *software* requisitos funcionais referem-se a uma função de um sistema e os seus componentes. Onde a função é descrita pelos seus componentes de entrada, saída e o seu comportamento. A descrição tem de obedecer a um balanço entre uma especificação demasiado abstrata e demasiado detalhada. Logo, descrições que possam ser demasiado vagas e pouco descritivas e não ajudaria tecnicamente ou descrições demasiado detalhadas que dificultaria na adaptação dos componentes em situações pontuais no decurso de um processo (Humphrey, 1989).

### 5.2.2. Funcionalidade de um Sistema Estrada Segura em Portugal

O Sistema Estrada Segura pretende melhorar a segurança dos automobilistas ao avisar em tempo real da proximidade de alguma situação que possa ter ocorrido, ou estar a decorrer, na via. A questão que surgiu foi: Como conseguir obter estas informações?

O desenvolvimento da aplicação Sistema Estrada Segura envolveu a consulta de um leque de informação que irei descrever de seguida.

- Quem o avisa<sup>25</sup> (PSP) – Mensalmente a PSP disponibiliza os locais dos radares.
- Radares fixos<sup>26</sup> – Lista obtida em Outubro de 2013. Neste momento só se encontra disponível para utilizadores registados.
- Mapas da Google – Mapas que possibilitam a utilização da aplicação em diferentes países.
- Lusoponte<sup>27</sup> - integrar em tempo real o vídeo das camaras da Ponte 25 de Abril e Ponte Vasco da Gama.
- Entrevista com Diretor das Estradas de Portugal sobre a viabilização de integrar as camaras na aplicação Estrada Segura<sup>28</sup>.
- Entrevistas para perceber o que os utilizadores procuram quando utilizam redes sociais para obter alertas sobre operações stop e problemas de tráfego.

### 5.2.3. Recolha de informação

Os utilizadores da aplicação inserem marcos a alertar de eventos na via. Também validam marcos que já se encontravam previamente inseridos: um utilizador recebe um alerta de acidente, imediatamente o seu nível de atenção altera, e

---

<sup>25</sup> <http://www.psp.pt/>

<sup>26</sup> <http://radaresdeportugal.pt/site/bases-de-dados-pdi/viewcategory/4-radares-atualizados>

<sup>27</sup> <https://www.lusoponte.pt/livecam.asp>

<sup>28</sup> <http://www.estradas.pt/index>

posteriormente pode validar o marco como um alerta correto. No caso contrário também poderá indicar na aplicação que aquele marco poderá ser removido. O que é importante?

- A interface da aplicação é tangível a qualquer automobilista? O desenho da aplicação é intuitivo?
- Em tempo real, é possível receber (no servidor) o alerta de novos marcos sem longos períodos de espera? Que informação será mais adequada para enviar?
- Como gerir marcos muito próximos? Elevada probabilidade de ser o mesmo alerta fornecido por automobilistas diferentes.
- É possível garantir a sincronização dos marcos nos vários dispositivos móveis?
- Como calcular eficientemente, em tempo real, a distância dos automobilistas aos marcos existentes?
- De que forma se deve utilizar a informação fornecida pelos automobilistas? Deveremos restringir esta informação apenas à aplicação?

Sendo os pontos acima questões importantes para o desenvolvimento da aplicação, seguidamente serão descritas as formas de recolha de informação utilizadas:

- Marcos inseridos pelos automobilistas.
- Validação (positiva e negativa) dos marcos.
- Utilizadores ativos (loja da Google *play store*).
- Número de instalações da aplicação (ao longo do tempo).
- Utilizadores ativos nos grupos do Facebook.
- Atividade nos grupos do Facebook: da aplicação ou apenas na rede social.

#### **5.2.4. Integração com redes sociais - Facebook**

A relevância e impacto das redes sociais foi uma questão importante que surgiu no processo de investigação. As redes sociais são um meio de grande sucesso e que ajudam no processo de divulgação de informação e, por isso, importante para o caso em estudo.

Na rede social Facebook existe uma comunidade que partilha ativamente alertas de congestionamento de vias causado por diferentes motivos (fluxo elevado de veículos, acidente, obras, etc) e, também alertam em relação a radares e operações stop. Estes alertas são inseridos manualmente, onde habitualmente apenas se encontra associado a uma breve descrição preenchida pelo utilizador. Em comparação com a aplicação proposta, Facebook tem a vantagem de se aproximar de um número mais elevado de pessoas e, tem a desvantagem de se encontrar afastado do automobilista. Também tem a desvantagem de não apresentar um mapa mais preciso de onde decorre a obstrução na via.

Com base neste cenário e, com o objetivo de complementar a aplicação proposta com as redes sociais, foi implementado um módulo que integra a aplicação Estrada Segura com a rede social Facebook. Neste módulo o objetivo será comunicar ao servidor aplicacional um novo marco e também no Facebook no grupo do Distrito associado ao marco.

### **5.3. Casos de uso**

Um requisito funcional poderá ser um cálculo, detalhes técnicos, manipulação de dados etc. Os requisitos comportamentais, que contemplam todos os casos do sistema, são extraídos dos casos de uso. Portanto um caso de uso corresponde a uma unidade funcional onde é descrita uma sequência de mensagens entre o sistema e outros componentes (e.g. dispositivo móvel ou facebook).

Os casos de uso são uma sequência de ações que um ou mais ator realizam de modo a obter um determinado resultado. Os diagramas de caso de uso permitem representar as interações entre os utilizadores e os sistemas e também as funcionalidades a implementar no sistema. É importante notar que os diagramas de casos de uso não descrevem o modo como se deve construir uma aplicação, mas sim, o modo como se deve comportar. De seguida é apresentado o diagrama de caso de uso (Figura 14) do sistema. No diagrama “Utilizador” representa um ator da aplicação. O “Utilizador” será o principal componente pois toda a interação do sistema através da interface móvel é feita por este ator.

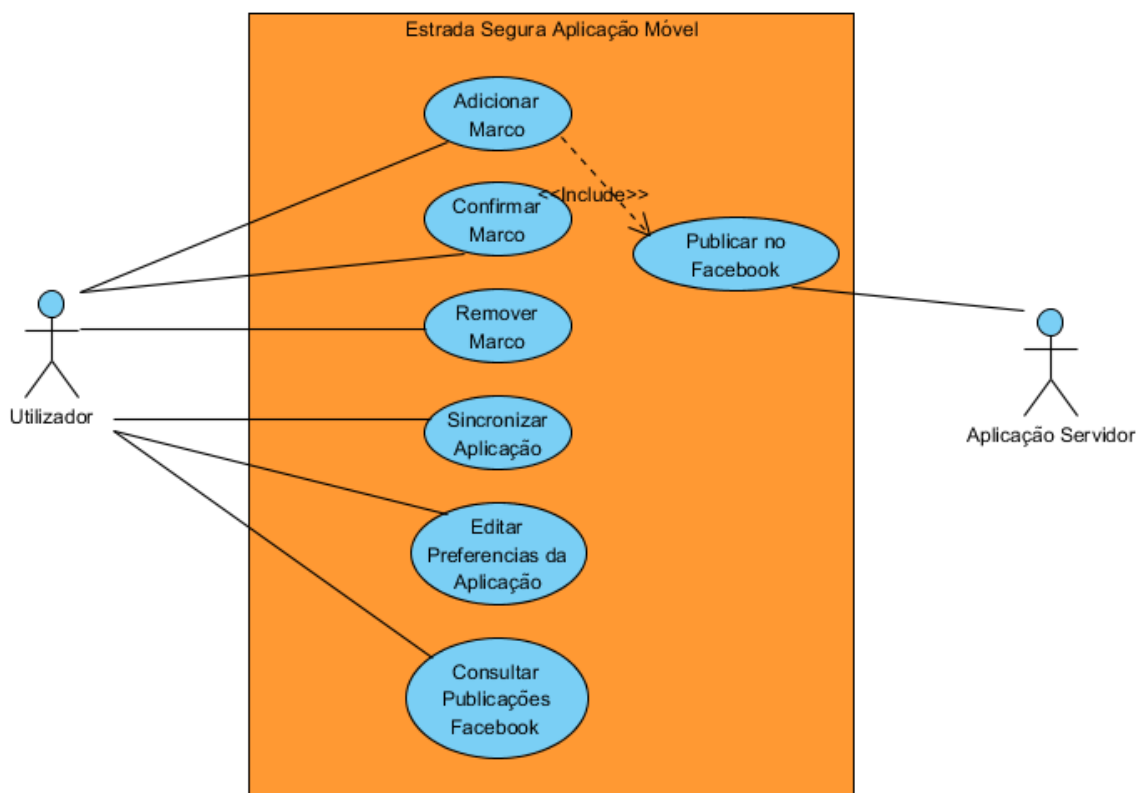


Figura 14 - Casos de uso da aplicação “Estrada Segura”.

De seguida será apresentado a descrição específica dos casos de uso mais relevantes apresentados no diagrama da Figura 14.

Tabela I - Caso de Uso - Adicionar Marco.


<b>Número</b>	<b>Nome do Caso de Uso</b>
1	Adicionar Marco
<b>Descrição</b>	
Este caso de uso descreve como o Utilizador adiciona um marco.	
<b>Atores</b>	
Utilizador	
<b>Pré Condições</b>	
A aplicação móvel tem de estar em funcionamento O dispositivo móvel tem de estar com serviço de ligação de dados ativo (wifi ou dados móveis) O sistema de coordenadas (GPS), do dispositivo móvel, tem de estar ativo	
<b>Sequência Normal de Ações</b>	
Pressiona o botão adicionar marco  Preenche o formulário apresentado (campos do formulário: tipo de marco, descrição) Envia o formulário	
<b>Sequência Alternativa (1)</b>	
Pressiona o ecrã durante aproximadamente 2 segundos, no local onde é pretendido adicionar um marco. A execução continua no ponto 2 da sequência normal	
<b>Sequência Alternativa (2)</b>	
Seleciona o menu da aplicação Pressiona o botão adicionar marco A execução continua no ponto 2 da sequência normal	



Tabela II - Caso de Uso - Confirmar Marco.

<b>Número</b>	<b>Nome do Caso de Uso</b>
1	Confirmar Marco
<b>Descrição</b>	
Este caso de uso descreve como o Utilizador confirma a existência de um marco.	
<b>Atores</b>	
Utilizador	
<b>Pré Condições</b>	
A aplicação móvel tem de estar em funcionamento O dispositivo móvel tem de estar com serviço de ligação de dados ativo (wifi ou dados móveis)	
<b>Sequência Normal de Ações</b>	
Seleciona o marco desejado Pressiona de cima da descrição Confirma a existência do marco pressionando o botão “confirmar”	

Tabela III - Caso de Uso - Remover Marco.

<b>Número</b>	<b>Nome do Caso de Uso</b>
1	Remover Marco
<b>Descrição</b>	
Este caso de uso descreve como o Utilizador pode remover um marco.	
<b>Atores</b>	
Utilizador	
<b>Pré Condições</b>	
A aplicação móvel tem de estar em funcionamento O dispositivo móvel tem de estar com serviço de ligação de dados ativo (wifi ou dados móveis)	
<b>Sequência Normal de Ações</b>	
Seleciona o marco desejado Pressiona de cima da descrição Remove o marco pressionando o botão “apagar”	

Tabela IV - Caso de Uso - Sincronizar Aplicação.



<b>Número</b>	<b>Nome do Caso de Uso</b>
1	Sincronizar Aplicação
<b>Descrição</b>	
Este caso de uso descreve como o Utilizador pode sincronizar a aplicação.	
<b>Atores</b>	
Utilizador	
<b>Pré Condições</b>	
A aplicação móvel tem de estar em funcionamento O dispositivo móvel tem de estar com serviço de ligação de dados ativo (wifi ou dados móveis)	
<b>Sequência Normal de Ações</b>	
Pressiona o botão sincronizar  Assim que o símbolo de sincronização  desaparecer a aplicação encontra-se atualizada	

Tabela V - Caso de Uso - Editar Preferências.

<b>Número</b>	<b>Nome do Caso de Uso</b>
1	Editar Preferências
<b>Descrição</b>	
Este caso de uso descreve como o Utilizador pode editar as preferências da aplicação.	
<b>Atores</b>	
Utilizador	
<b>Pré Condições</b>	
A aplicação móvel tem de estar em funcionamento.	
<b>Sequência Normal de Ações</b>	
Seleciona o menu da aplicação. Pressiona o botão de “editar preferências”. Edita a preferência desejada. Pressiona o botão “guardar”.	

Tabela VI - Caso de Uso - Consultar Publicações Facebook.


Número	Nome do Caso de Uso
1	Consultar Publicações Facebook
<b>Descrição</b>	
Este caso de uso descreve como o Utilizador pode consultar as publicações nos diferentes grupos do Facebook.	
<b>Atores</b>	
Utilizador	
<b>Pré Condições</b>	
A aplicação móvel tem de estar em funcionamento O dispositivo móvel tem de estar com serviço de ligação de dados ativo .	
<b>Sequência Normal de Ações</b>	
Pressiona o botão do Facebook  São listadas as publicações do grupo do Facebook correspondente ao distrito onde se encontra	
<b>Sequência Alternativa (1)</b>	
Este fluxo alternativo começa no ponto 2 do fluxo normal Seleciona o distrito desejado Continua no ponto 2 do fluxo normal	

Tabela VII - Caso de Uso - Publica no Facebook.

Número	Nome do Caso de Uso
1	Publicar no Facebook
<b>Descrição</b>	
Este caso de uso descreve como o Sistema publica um marco no Facebook.	
<b>Atores</b>	
Aplicação Servidor	
<b>Pré Condições</b>	
O servidor tem de ter ligação à internet ativa	
<b>Sequência Normal de Ações</b>	
Guarda os dados na base de dados Calcula o distrito referente às coordenadas do marco Envia a publicação para o Facebook	

## 5.4. Diagrama entidade relacionamentos

O modelo relacional – diagrama entidade relacionamentos – descreve os dados de um sistema e os seus requisitos. De uma forma abstrata informa qual a informação do domínio do sistema que será implementada na base de dados. Os componentes principais dos modelos ER (entidade relacionamentos) são as entidades (objetos) e as relações que existem entre as mesmas. Portanto num modelo entidade relacionamento é uma maneira sistemática de descrever e definir um processo de negócio. O processo é modelado com componentes (entidades) que são ligadas umas às outras por relacionamentos que expressam as dependências e exigências entre elas. E, o desenho do modelo ER não depende de uma aplicação informática (Chen, 1976).

Um modelo ER é normalmente implementado como uma base de dados .Nos casos de uma base de dados relacional, que armazena dados em tabelas, as próprias tabelas representam as entidades. Alguns campos de dados nestas tabelas apontam para índices em outras tabelas. Tais ponteiros representam relacionamentos que se caracterizam em três tipos distintos:

- Relacionamento 1..1 (relacionamento um para um) - indica que as tabelas têm relacionamento apenas entre si. Deve escolher qual tabela receberá a chave estrangeira;
- Relacionamento 1..n (relacionamento um para muitos) - a chave primária da tabela que tem o lado 1 vai para a tabela do lado N. No lado N ela é chamada de chave estrangeira;
- Relacionamento n..n (relacionamento muitos para muitos) - quando tabelas têm um relacionamento n..n é necessário criar uma nova tabela com as chaves primárias das tabelas envolvidas. E, é gerada uma chave composta, ou seja, formada por diversos campos de chaves primárias de outras tabelas. Com a chave composta o relacionamento será 1..n, sendo que o lado n ficará com a nova tabela criada.

De seguida apresenta-se os relacionamentos entre as entidades marco (MARKER), tipo de marco (MARKER\_TYPE), dispositivo móvel (DEVICE) e confirmação (MARKER\_CONFIRMATION).

- O tipo marco tem uma relação de 1..n com o marco. O marco contém uma chave estrangeira para o tipo de marco. Sendo que um marco só pode conter um tipo de marco, mas um tipo de marco pode estar presente em n marcos.
- O dispositivo móvel tem uma relação de 1..n com o marco. O marco contém uma chave estrangeira para o dispositivo móvel. Sendo que um marco só pode conter um dispositivo móvel, mas um dispositivo móvel pode estar presente em n marcos.

- O marco e o dispositivo móvel tem uma relação de 1..n com a confirmação. Por isso a confirmação do marco tem uma chave estrangeira tanto do marco como do dispositivo móvel. Neste caso pode-se dizer, através da confirmação, que existe um relação de n..n entre o marco e o dispositivo móvel.

O modelo apresentado na Figura 15, Figura 18 e Figura 16 apresentam as relações entre as tabelas da aplicação Estrada Segura. Para cada tabela é apresentado os campos que as compõem.

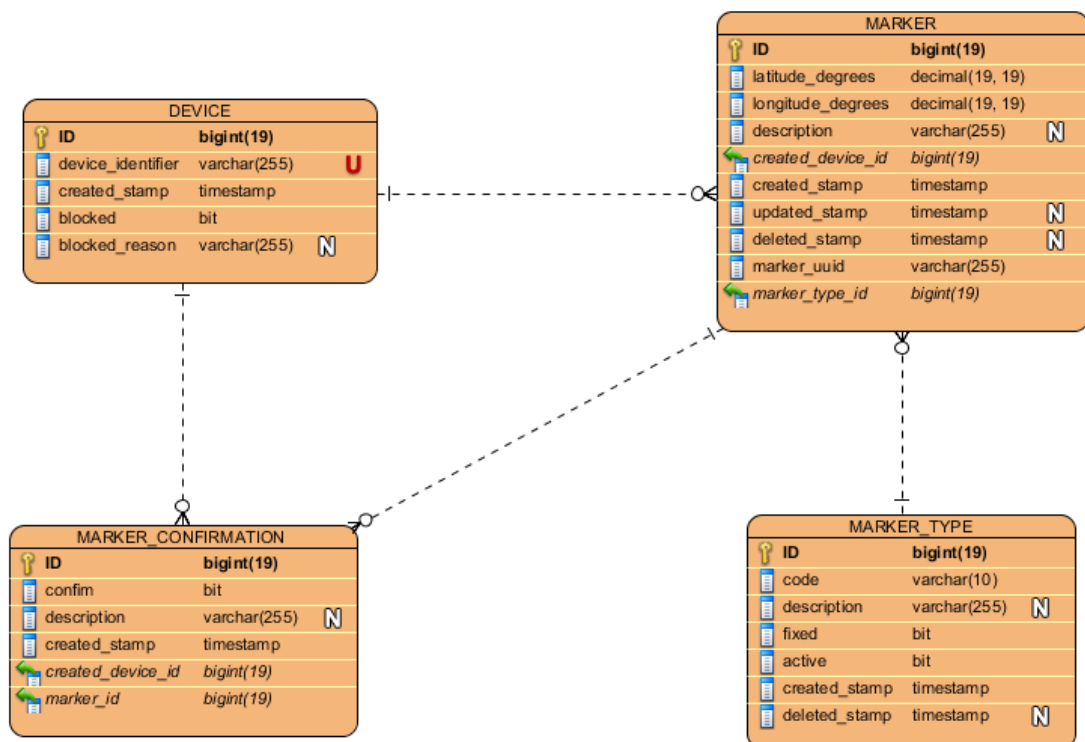


Figura 15 - Modelo ER servidor.

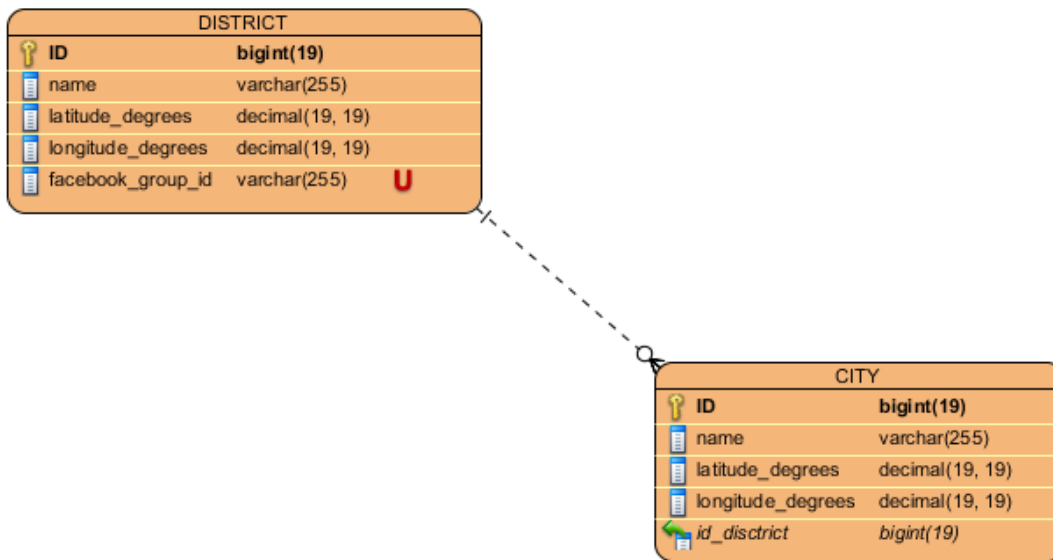


Figura 16 - Modelo ER integração com o Facebook.

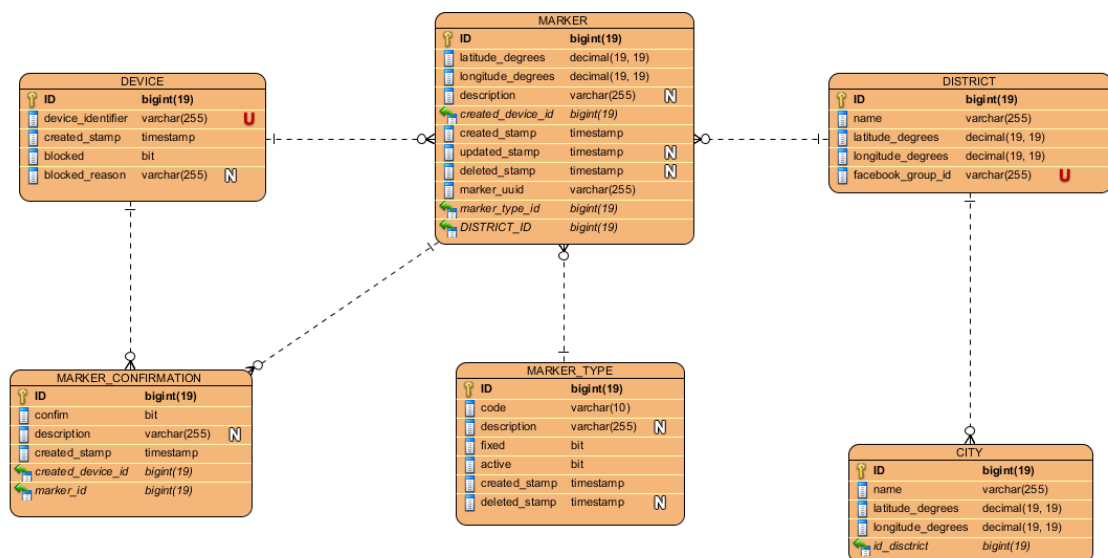


Figura 17 - Modelo ER completo após integração com o Facebook.














MARKER		
 ID	varchar(255)	
 snippet	varchar(255)	N
 latitude_degrees	decimal(19, 19)	
 longitude_degrees	decimal(19, 19)	
 creation_date	timestamp	
 synchronization_date	timestamp	N
 type	tinyint(3)	
 fixed	bit	
 created_by_me	bit	
 confirmations	integer(10)	
 last_confirmation	timestamp	
 denials	integer(10)	
 last_denial	timestamp	N

Figura 18 - Modelo ER dispositivos móveis.

## 5.5. Descrição de tabelas

Nos modelos de bases de dados relacionais, a tabela é um conjunto de dados dispostos em número infinito de colunas e número ilimitado de linhas. As colunas são tipicamente consideradas os campos da tabela, e caracterizam os tipos de dados que deverão constar na tabela (numéricos, alfanuméricos, datas, coordenadas, etc..). O número de linhas pode ser interpretado como o número de combinações de valores dos campos da tabela e pode conter linhas idênticas, dependendo do objetivo. A forma de referenciar inequivocamente uma única linha é através da utilização de uma chave primária. Para além do tipo de dados inerente a todas as colunas de uma tabela algumas podem ter associadas restrições. Por exemplo, a unicidade (SQL: UNIQUE), proibição de valores NULL (SQL: NOT NULL), delimitação de valores, etc. Estas restrições impedem que sejam inseridos valores não desejados que comprometam a validade e integridade dos dados.

O número de linhas de uma tabela é virtualmente ilimitado, o que torna as pesquisas por valor potencialmente muito lentas. Para permitir agilizar estas consultas, podem ser associados índices à tabela. Os índices são estruturas de dados independentes da forma e ordem como estão armazenados os dados embora tenham relação direta com os mesmos. Como consequência cada alteração de dados irá corresponder a uma (ou mais) alterações em cada um dos índices, aumentando o esforço necessário ao sistema gestor de base de dados (SGBD) para gerir essa alteração. E, por isso, os índices não existem para cada coluna. A estrutura usada para a elaboração do índice depende do SGBD e do tipo de dados das colunas usadas no índice.

Não obstante o papel principal da tabela ser a de armazenamento de dados, é também utilizada como representação de relações entre tabelas (entidades), tipicamente designadas por chaves estrangeiras. De seguida apresenta-se, para este caso de estudo, as tabelas e os seus atributos (Tabela VIII, Tabela IX, Tabela X, Tabela XI, Tabela XII, Tabela XIII e Tabela XIV). Para cada um dos atributos é apresentado o nome e o tipo. É também indicado se o seu preenchimento é obrigatório (NOT NULL), se é chave primária (PK) ou chave estrangeira (FK) e se é único (UNIQUE).

Tabela VIII - Dispositivo Móvel.

	<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Restrições</b>
PK	ID	inteiro	Chave primária da tabela DEVICE	NOT NULL
	device_identifier	texto	String que identifica o telefone/tablet. Designado por “Device ID”	NOT NULL UNIQUE
	created_stamp	data	Data de registo do equipamento no sistema	NOT NULL
	blocked	verdadeiro/falso	Se verdade indica que o equipamento se encontra bloqueado	NOT NULL
	blocked_reason	texto	Qual o motivo pelo qual o equipamento foi bloqueado. Preenchido quando se bloqueia o equipamento	



Tabela IX - MARKER\_TYPE.

	<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Restrições</b>
PK	ID	inteiro	Chave primária da tabela MARKER_TYPE	NOT NULL
	code	texto	Código que identifica o tipo de marco	NOT NULL
	description	texto	Descrição do tipo de marco	NOT NULL
	fixed	verdadeiro / falso	Se verdade é um marco fixo, se falso é um marco móvel	NOT NULL
	active	verdadeiro / falso	Se o tipo de marco se encontra ativo	NOT NULL
	created_stamp	data	Data de inserção do registo	NOT NULL
	deleted_stamp	data	Data de remoção do registo	

Tabela X - MARKER\_CONFIRMATION.

	<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Restrições</b>
PK	ID	inteiro	Chave primária da tabela MARKER_CONFIRMATION	NOT NULL
	confirm	verdadeiro / falso	Se verdade foi é uma confirmação. Se falso é uma remover do marco	NOT NULL
	description	texto	Descrição da confirmação/remoção	
FK	created_device_id	inteiro	ID do DEVICE que inseriu o registo	NOT NULL
FK	marker_id	inteiro	ID do marco que foi confirmado/removido	NOT NULL
	created_stamp	data	Data de inserção do registo	NOT NULL

Tabela XI - MARKER (Server).

	<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Restrições</b>
PK	ID	inteiro	Chave primária da tabela MARKER	NOT NULL
	latitude_degrees	decimal	Latitude em graus	NOT NULL
	longitude_degrees	decimal	Longitude em graus	NOT NULL
	description	texto	Descrição do marco	
FK	created_device_id	inteiro	Chave estrangeira para a tabela DEVICE	NOT NULL
	created_stamp	data	Data de inserção do registo	NOT NULL
	updated_stamp	data	Data de alteração do registo	
	deleted_stamp	data	Data de remoção do registo	
	marker_uuid	texto	Identificador único partilhado entre os diferentes dispositivos e SGBD's	NOT NULL UNIQUE
FK	marker_type_id	inteiro	Chave estrangeira para a tabela MARKER_TYPE	NOT NULL
FK	district_id	inteiro	Chave estrangeira para a tabela DISTRICT	

Tabela XII - DISTRICT (Server).

	<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Restrições</b>
PK	ID	inteiro	Chave primária da tabela DISTRICT	NOT NULL
	name	texto	Nome do distrito	
	latitude_degrees	decimal	Latitude em graus	NOT NULL
	longitude_degrees	decimal	Longitude em graus	NOT NULL
	facebook_group_id	texto	ID do grupo do facebook a que corresponde o distrito	NOT NULL UNIQUE

Tabela XIII - CITY (Server).

	<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Restrições</b>
PK	ID	inteiro	Chave primária da tabela CITY	NOT NULL
	Name	texto	Nome da localidade	
	latitude_degrees	decimal	Latitude em graus	NOT NULL
	longitude_degrees	decimal	Longitude em graus	NOT NULL
	id_district	inteiro	Chave estrangeira para a tabela distrito	NOT NULL

Tabela XIV - MARKER (Dispositivo móvel).

	<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>	<b>Restrições</b>
PK	ID	texto	Chave primária da tabela MARKER (UUID)	NOT NULL
	snippet	texto	Descrição do marco	
	latitude_degrees	decimal	Latitude em graus	NOT NULL
	longitude_degrees	decimal	Longitude em graus	NOT NULL
	creation_date	data	Data de inserção do registo	NOT NULL
	synchronization_date	data	Data de sincronização do registo	NOT NULL
	type	inteiro	Identificação do tipo de marco	NOT NULL
	fixed	verdadeiro/falso	Se verdade é um marco fixo, se falso é um marco móvel	NOT NULL
	created_by_me	verdadeiro/falso	Verdadeiro foi criado pelo device, falso foi criado por outro device	NOT NULL
	confirmations	inteiro	Numero total de confirmações	NOT NULL
	last_confirmation	data	Data da última confirmação	NOT NULL
	Denials	inteiro	Numero total de remoções do marco	NOT NULL
	last_denial	data	Data da última remoção	NOT NULL



## 6. Estrada Segura

Como já tem sido discutido nos capítulos anteriores a aplicação Estrada Segura tem como objetivo fornecer uma ferramenta onde os automobilistas possam ter a informação em tempo-real de possíveis perigos na estrada. Esta aplicação integra marcos fixos (e.g. Radares) e marcos inseridos pelos utilizadores (automobilistas). Sendo este segundo tipo de marcos os mais relevantes pois um utilizador da aplicação poderá visualizar um problema na via antes desta se encontrar congestionada, evitando possíveis agravamentos ao problema e possibilitando que outros automobilistas possam escolher entre duas opções: circular com maior precaução ou escolher, atempadamente, outra via. Um dos problemas onde foi proposta uma solução no desenvolvimento da aplicação foi no cálculo da distância entre marcos e o como calcular o distrito que um automobilista se encontra.

### 6.1. Distância entre marcos

No cálculo de distâncias entre diferentes pontos no mapa o desafio será em como realizar esta tarefa de forma eficiente. Portanto se existir um número elevado de pontos no mapa como realizar uma pesquisa à base de dados e obter a resposta de forma a não sobrecarregar o dispositivo (Matuscheck, 2011).

Cada marco é representado pela sua posição em termos de latitude e longitude. E, de acordo com o que é descrito no trabalho de (Matuscheck, 2011) a forma que não se deve seguir seria computar o ponto mínimo e máximo de longitude ao longo de um círculo de raio R. Isto porque existem pontos que não se encontram dentro do mesmo círculo de raio R mas estão próximos do ponto (marco) e, conseqüentemente, são perdidos através deste método. O autor sugere que a forma de ultrapassar este problema será calcular a proximidade entre dois pontos através de:

$$lat_T = \arcsin\left(\frac{\sin(lat)}{\cos(r)}\right), \quad (1)$$

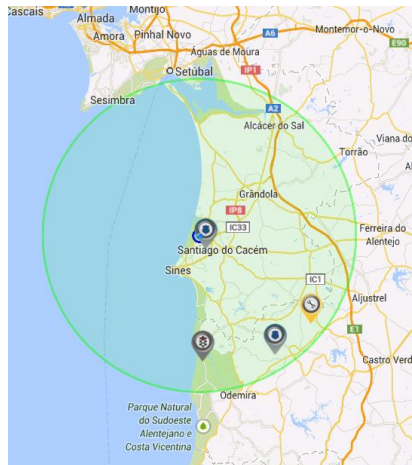
$$lon_{min} = lon_{T_1} = lon - \Delta lon, \quad (2)$$

$$lon_{max} = lon_{T_2} = lon + \Delta lon, \quad (3)$$

$$\Delta lon = \frac{\arccos\left(\frac{\cos(r) - \sin(lat_T)}{\cos(lat_T) \cdot \cos(lat)}\right)}{\arcsin\left(\frac{\sin(r)}{\cos(lat)}\right)}. \quad (4)$$

Nas equações (1), (2), (3) e (4)  $T_1$  e  $T_2$  correspondem a dois marcos e  $r$  ao raio do círculo. Portanto os pontos são obtidos através do dispositivo móvel e contem as coordenadas em termos de latitude e longitude. A decisão seria definir qual seria o raio máximo que se pretende alertar o automobilista. E foi definido o raio de 3 quilómetros, como esta distância foi definida com base no senso comum e, por isso, é muito subjetiva foi realizado um questionário onde se perguntou aos participantes se o momento em que recebiam um alerta é adequado (Q13 **Tabela XVII**), no próximo capítulo este estudo é discutido com maior pormenor.

Na aplicação existem em simultâneo vários marcos na base de dados e, por isso, é deveras importante realizar uma chamada à base de dados que retorne os marcos que se encontram dentro do raio de forma eficiente. A chamada à base de dados adotada para esta tarefa encontra-se otimizada para filtrar os marcos que não pertencem ao círculo em torno de um dado automobilista (Figura 19 e Figura 20):

$$\begin{aligned} &SELECT * FROM ( SELECT * FROM Places WHERE (Lat \geq 1.2393 AND Lat \leq 1.5532) AND (Lon \geq -1.8184 AND Lon \leq 0.4221) AND a \cos(\sin 1.3963) AND \sin(Lat) + \cos(Lon - (-0.6981)) \leq 0.1570. \end{aligned} \quad (5)$$


**Figura 19 - Raio de sincronização de dados**

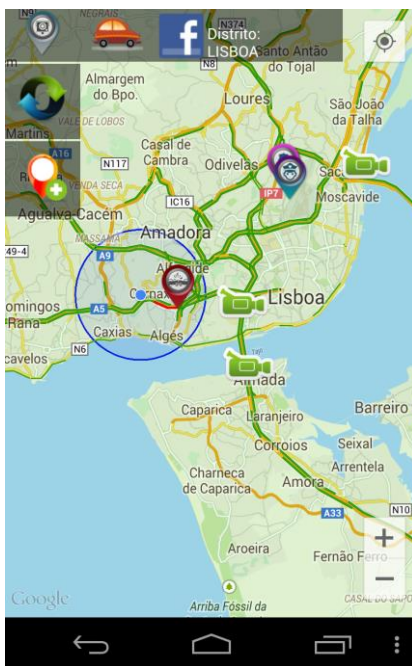


Figura 20 - Aplicação móvel com um acidente dentro do raio de alertas

## 6.2. Cálculo do distrito

A integração da aplicação Estrada Segura com o Facebook gerou a seguinte questão: a existência de apenas um grupo para os alertas direcionados aos automobilistas de Portugal tem como consequência que as zonas mais populosas controlavam o grupo. Por exemplo, um automobilista da zona de Lisboa terá maior facilidade de encontrar avisos da sua zona pois trata-se de uma zona maior onde existe uma concentração maior de tráfego. A solução proposta para informar os utilizadores nas diferentes zonas do país foi criar um grupo por cada distrito de Portugal (Tabela XV). Portugal continental tem 18 distritos: Aveiro, Beja, Braga, Bragança, Castelo Branco, Coimbra, Évora, Faro, Guarda, Leiria, Lisboa, Portalegre, Porto, Santarém, Setúbal, Viana do Castelo, Vila Real e Viseu (Figura 21).

Tabela XV - Número de utilizadores por distrito no Facebook.

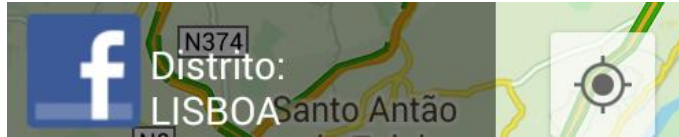
<b>Grupo Facebook</b>	<b>Utilizadores</b>	<b>Grupo Facebook</b>	<b>Utilizadores</b>
Geral	14.423	Porto	333
Leiria	5.042	Santarém	283
Castelo Branco	4.090	Setúbal	273
Coimbra	2.591	Braga	241
Lisboa	1.543	Portalegre	133
Aveiro	749	Madeira	118
Vila Real	638	Faro	113
Viseu	427	Évora	109
Guarda	422	Açores	72
Viana do Castelo	377	Bragança	67

O desafio será como eficientemente calcular a que distrito um dado ponto pertence. A computação necessária para calcular a distância entre dois pontos (6.1





mapa sem se encontrar nessa localização. O utilizador pode ser um automobilista aquando do momento de inserir o alerta ou ter tido conhecimento de uma ocorrência e inserir o alerta na aplicação.



**Figura 22** - Botão para o grupo do Facebook correspondente ao distrito da minha localização.

A autarquia digital disponibiliza a informação dos pontos que representam a fronteira dos polígonos dos distritos de Portugal continental<sup>29</sup>. E, o cálculo para determinar se um dado ponto pertence a um distrito é descrito no pseudo-código “Cálculo do Distrito”<sup>30</sup>. Para o cálculo apresentado no pseudo-código foi aplicado o seguinte raciocínio: é possível decidir a orientação entre dois vetores através do produto vetorial, e consoante o seu sentido pode-se identificar a respetiva orientação (horária ou anti-horária em relação a um ponto  $p$ ), onde  $p$  representa o ponto a que se pretende determinar o distrito. E  $P$  o respetivo polígono. Logo considerando três pontos  $p, a$  e  $b$  onde  $a$  e  $b$  perfazem uma semi-reta onde  $p$  intercepta e está numa posição vertical em relação a  $b$ . É possível calcular o produto vetorial para decidir a direção de  $p$  em relação ao segmento  $ab$ . Kunigami descreve todos os casos especiais que são detetados através deste método. Portanto para determinar o distrito de um ponto será necessário realizar vários cálculos matemáticos, reforçando a necessidade de realizar esta tarefa no servidor e de forma eficiente.

---

<sup>29</sup> <http://autarquiadigital.com/2013/07/27/carta-administrativa-oficial-de-portugal-caop-2013/>.

<sup>30</sup> Descrito por Kunigami em <https://kuniga.wordpress.com/2010/11/19/ponto-dentro-de-poligono/>.

### Pseudo-código: Cálculo do Distrito

---

1.  $cruzamentos \leftarrow 0;$
  2. *Para cada segmento  $p_i p_{i+1}$ :*
  3.  $a \leftarrow p_i;$
  4.  $b \leftarrow p_{i+1};$
  5.  $r \leftarrow (x_a - x_b)(y_p - y_b) - (x_p - x_b)(y_a - y_b);$
  6. *Se  $r = 0$  e  $\overrightarrow{pa} \cdot \overrightarrow{pb} \leq 0$ , então  $p$  está na fronteira de  $P$ ;*
  7. *Se  $y_a > y_b$ , troque  $a$  com  $b$ ;*
  8. *Se  $r < 0$  e  $y_a < y_p < y_b$ , então  $cruzamentos += 1$*
  9. *Para.*
  10. *Se  $cruzamentos$  é par,  $p$  está fora de  $P$ ;*
  11. *Se não,  $p$  está dentro de  $P$ ;*
-

### 6.3. Estrada Segura na loja play (Google)

A aplicação Estrada Segura<sup>31</sup> foi lançada na loja play do Google em Outubro de 2013. Na primeira versão da aplicação ainda não existia na interface da aplicação os botões de acesso rápido (Figura 27). Esta foi a primeira versão da aplicação e prévia ao estudo de usabilidade – descrito com maior pormenor no próximo capítulo. Como é possível observar na Figura 23 desde Outubro de 2013 a Janeiro de 2014 6.395 utilizadores descarregaram a aplicação. O que perfaz um pouco mais de 70 instalações diárias. Considerando que a aplicação não usufruiu de suporte publicitário para promover a mesma, estes números representam a necessidade de existir uma aplicação que consiga eficientemente disponibilizar este tipo de alertas.

Na Figura 24 é observado o dia onde foi realizada a integração da rede social Facebook. Em aproximadamente um ano (Janeiro de 2014 a Dezembro de 2014) o ritmo crescimento de descarregamentos da aplicação desceu bastante. Durante este período houve um crescimento de cerca de 21%. Contudo se o ritmo de crescimento fosse aproximado ao observado nos primeiros 3 meses dever-se-ia estar a observar um número próximo de 25.000 utilizadores. Neste ponto é preciso considerar que quando surgem novas aplicações e que apresentam características diferenciadoras do que existe no mercado existe um crescimento mais elevado e, posteriormente observa-se descida do ritmo de crescimento. No entanto, 8.109 utilizadores representa um número muito baixo para o número de utilizadores que seria expectável de necessitarem desta aplicação. Logo, a integração com a rede social Facebook tem como objetivo aumentar o nível de satisfação dos utilizadores, ao integrar uma componente que atualmente é muito valorizada, e divulgar em uma rede social que contém um número muito elevado de utilizadores a aplicação Estrada Segura.

Na Figura 25 é apresentada a evolução da aplicação Estrada Segura desde o seu aparecimento até Fevereiro de 2015. É de notar que desde a integração com o Facebook em Dezembro de 2014 e Fevereiro de 2015, aproximadamente 3 meses, houve um crescimento de 34%. Portanto é visível o impacto positivo quando se utilizam redes sociais para divulgar uma aplicação que dependente da intervenção dos utilizadores para divulgarem informação.

---

<sup>31</sup> A aplicação foi lançada com o nome de Radares da Polícia.

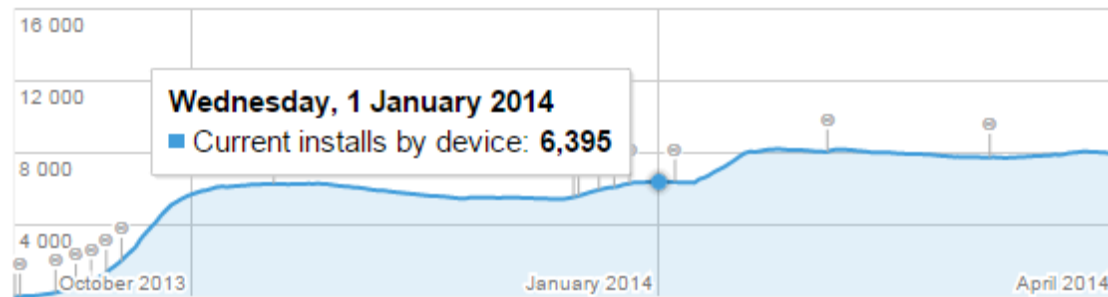


Figura 23 - Evolução da aplicação desde Outubro 2013 a 1 Janeiro de 2014.

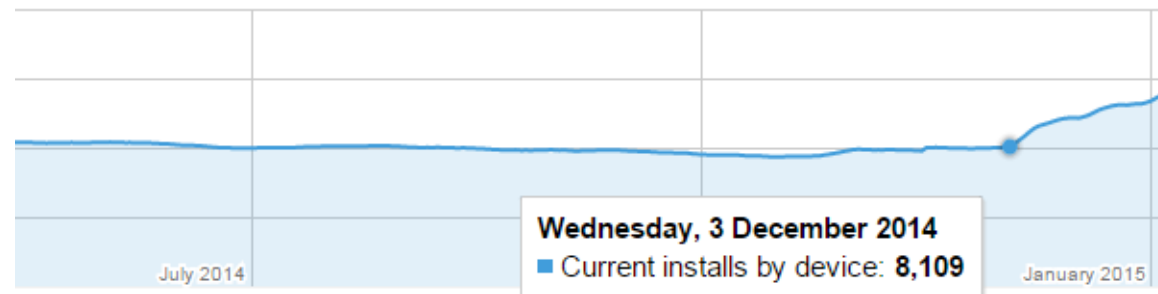


Figura 24 - Integração da rede social Facebook na aplicação (3 Dezembro 2014).

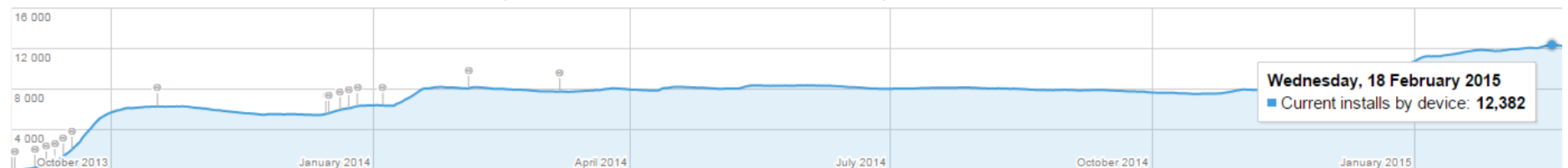


Figura 25 - Descarregamentos da aplicação Estrada Segura na loja play desde Outubro de 2013 a Fevereiro 2015.



## 7. Usabilidade da Interface

Neste capítulo será discutido a usabilidade da aplicação proposta. A medida de usabilidade pode ser descrita como uma medida que avalia a perspectiva do utilizador face à facilidade de utilização e utilidade de um sistema. Usualmente muito discutido nas áreas de HCI (*Human Computer Interaction*) e UX (*User experience*) (Butler, Jacob, & John, 1998; Choi & Lee, 2012; Dumas & Redish, 1993).

Os testes de usabilidade envolvem participantes que estão a executar tarefas predefinidas. O feedback recebido é avaliado através de diferentes métodos e, é utilizado para realizar correções aos erros do desenho da aplicação, corrigir problemas etc. Existe uma necessidade de aumentar o número de testes feitos no mundo-real, isto porque atualmente maior parte das aplicações que utilizam informação de localização são maioritariamente testadas em laboratório e é raro ser feito um testes no campo (+- 19%) (Duh, Tan, & Chen, 2006; Kjeldskov & Graham, 2003).

De acordo com a ISO 9241-11 (Standardization, 1998) usabilidade pode ser dividida em três parâmetros fundamentais: eficácia, eficiência e satisfação do utilizador<sup>32</sup> (Jordan, 1998). Estes parâmetros são fundamentais para uma aplicação que utiliza localização geográfica e, testes de usabilidade são um aspeto importante para avaliar cada um dos parâmetros in condições reais. Estas normas devem ser tidas em consideração quando se realizam os testes de usabilidade, ou seja, durante a fase de desenho e desenvolvimento da aplicação (Capítulo 0 – Figura 13). A falta destes testes poderá condenar uma aplicação ao seu fracasso. Uma aplicação que o seu sucesso depende dos utilizadores e provoca desagrado e rejeição, pelos respetivos, muito dificilmente será adotada por vários utilizadores (Kimber, Georgievski, & Sharda, 2005). No trabalho de (Nivala, 2005) é descrito um conjunto de elementos que expandem os três fundamentais em seis:

- Eficácia - a capacidade dos utilizadores conseguirem realizar as diferentes tarefas e satisfazer os objetivos pretendidos de forma assertiva e correta.
- Eficiência – considerando o tempo ideal que será necessário para realizar cada tarefa qual foi a performance dos utilizadores a realizar a mesma.
- Satisfação – o nível de contentamento do utilizador ao utilizar as funcionalidades da aplicação.
- Aprendizagem – o nível de dificuldade de aprender a utilizar a aplicação.

---

<sup>32</sup> Do inglês: *effectiveness*, *efficiency* e *user satisfaction*.

- Memória – capacidade de lembrar como se utiliza a aplicação após um período sem utilizar a mesma.
- Erros – o número de erros ocorridos durante o uso da aplicação e como superaram os mesmos.

Satisfação e memória são elementos que podem ser utilizados para avaliar a eficácia e eficiência, pois um produto que seja tenha estas características mais facilmente é aprendido e recorda-se de como o utilizar. Analogamente, erros também se encontra corelacionado com eficácia. Logo usabilidade é principalmente avaliado por os três elementos principais: eficácia, eficiência e satisfação.

Na área de HCI os testes de usabilidade têm maior foco nas interfaces de sistemas (Helander, Landauer, & Prabhu, 1997). E existem várias técnicas que são utilizadas para avaliar a usabilidade: observação de como os utilizadores utilizam a aplicação; entrevista onde existe uma comunicação sobre a experiência que os utilizadores tiveram/estão a ter com a aplicação; um questionário onde são realizadas um conjunto de questões para avaliar a aplicação entre outras. Para aplicações de PC existem vários estudos para avaliar a usabilidade através técnicas baseadas em HCI, no entanto ainda não existe uma metodologia pré-estabelecida para aplicações móveis (Alshehri & Freeman, 2012). Um dos fatores que torna esta tarefa mais complicada é o fato de não ser previsível o ambiente no mundo real onde o utilizador irá utilizar a aplicação (e.g. conduzir, estacionar, parque, etc.).

O teste de usabilidade foi realizado com 10 participantes (Tabela XVI). Foram selecionados participantes com diferentes níveis de experiência com tecnologias. Para a avaliação não foi fornecida nenhuma informação prévia, apenas o nome da aplicação e onde esta se encontrava. E, foram dadas as seguintes tarefas:

1. Abrir a aplicação e visualizar as diferentes opções (5 minutos).
2. Realizar um conjunto de tarefas (10 minutos):
  - a. Identificar onde se encontra ajuda para utilizar a aplicação. E informação sobre a mesma.
  - b. Adicionar um marco.
  - c. Visualizar a descrição de um marco pré-existente.
  - d. Confirmar ou apagar um marco.
3. Responder a um questionário (5 minutos).


O estudo de utilizador é realizado após o utilizador terminar as várias tarefas exploratórias e um questionário escrito. Como é possível visualizar na Tabela XVII o questionário está dividido em quatro partes, cada uma com quatro questões. A primeira parte visa avaliar a usabilidade da interface, a segunda parte avalia se o



utilizador concorda com os componentes disponíveis na aplicação, a terceira parte tem como objetivo avaliar a dificuldade em efetuar diferentes operações com a aplicação e, na quarta parte o objetivo será avaliar a sincronização e o tempo que decorre para realizar diferentes tarefas. Para a primeira parte ir-se-á avaliar os elementos de usabilidade: satisfação, memória e aprendizagem; na segunda parte: satisfação; na terceira parte: eficácia, eficiência, aprendizagem e erros; e na quarta parte: eficiência e erros. O utilizador classifica cada uma das questões com uma pontuação de 1 a 5 (*5-point-Likert-scale*).

**Tabela XVI - Descrição dos participantes.**

<b>Participant e</b>	<b>Área Profissional</b>	<b>Idade</b>	<b>Sexo</b>	<b>Experiência com Tecnologias (1-5)</b>
<b>1</b>	Engenharia de Informática	31	F	5
<b>2</b>	Engenharia Mecânica	30	F	4
<b>3</b>	Matemática	35	F	3
<b>4</b>	Economia	28	M	4
<b>5</b>	Engenharia Civil	31	M	4
<b>6</b>	Enfermeiro	32	M	2
<b>7</b>	Web Design	33	M	5
<b>8</b>	Distribuição de Encomendas	25	M	2
<b>9</b>	PSP - Polícia de Segurança Pública	29	M	3
<b>10</b>	Fotografia	23	F	2

Em geral os resultados obtidos foram bastante bons mas existem algumas questões que apresentaram um grau de satisfação maior que outras. A média do nível de satisfação por cada questão está apresentado na penúltima coluna da Tabela XVII. É possível observar que a questão Q3 foi a que apresentou o desvio padrão mais elevado e o nível de satisfação mais baixo – média de 3.2. Esta questão encontra-se relacionada com a posição dos botões, onde o botão mais utilizado é o marco – este botão  possibilita adicionar um novo marco. A questão que teve o nível mais alto de satisfação e desvio padrão mais baixo foi a Q14.

Em relação à primeira parte do estudo de usabilidade os participantes demonstraram desagrado com a localização dos botões as apresentaram um nível de satisfação muito elevado para a qualidade da visualização: ícones, identificação dos marcos e letra utilizada para apresentar informação textual. A média na pontuação da escala likert varia entre 4.5 e 4.1 onde o desvio padrão apenas 0.5 e 0.54. Somente será necessário ter em atenção a Q3, ainda neste capítulo irei discutir como esta questão foi resolvida.

Relativamente aos componentes disponíveis (segunda parte do estudo de usabilidade) o nível de satisfação variou entre 4.3 e 4.6 com um desvio padrão médio de 0.5. E por isso depreende-se que os participantes se encontravam satisfeitos com os diferentes componentes. Por outras palavras, na opinião deste grupo de participantes a aplicação não apresenta funcionalidades redundantes ou desnecessárias.

A terceira parte do estudo focou a capacidade de aprendizagem e eficiência a realizar diferentes operações. Esta parte tem uma importância muito elevada, se um participante não consegue realizar uma tarefa, ou tem muita dificuldade a realizar a mesma, irá provocar um sentimento de frustração e conseqüentemente desagrado com a aplicação. Neste grupo a pontuação variou entre 4.2 e 4.8 com um desvio padrão médio ligeiramente inferior ao observado na segunda parte do estudo de usabilidade.

Na quarta e última parte do estudo de usabilidade o objetivo seria observar os tempos que a aplicação demora a realizar diferentes tarefas e qual é a percepção dos participantes perante esse tempo de espera. E, este grupo foi o que apresentou o melhor nível de satisfação. Nos capítulos 2 e 0 é discutido a importância de sincronização e a importância de distribuir tarefas sobre o dispositivo móvel e o servidor. O resultado obtido pelos participantes são bastante satisfatórios, pois comprovam a importância de estudar os recursos necessários para cada tarefa, distribuir pelo dispositivo móvel e servidor, evitar comunicações redundantes, entre outros aspetos que foram discutidos nos capítulos mencionados. Outro aspeto importante é como a aplicação tem a capacidade de sincronizar com o GPS ou obter

a triangulação das antenas para obter a posição do automobilista de forma eficiente (Q13) e em simultâneo, apesar das comunicações realizadas para sincronizar com os marcos existentes no servidor, o dispositivo móvel não aquece por estar a realizar demasiadas comunicações ou operações (e.g. cálculo da proximidade entre o automobilista e marcos existentes). Em relação à questão Q13 o automobilista tem a possibilidade de selecionar o raio de alerta de acordo com a distância que seja adequada ao seu nível de satisfação, o que poderá também ter aumentado o nível de satisfação em relação ao momento em que o automobilista recebe o alerta.






**Tabela XVII** - Questionário sobre a aplicação Estradas Segura. As respostas são avaliadas numa escala de Likert de 1 a 5, onde 1 - Não Concordo e 5 - Concordo. A média corresponde à média aritmética e desvio padrão é uma medida estatística que identifica a variação em relação à média.

Questões	Participantes										Média	Desvio Padrão
	1	2	3	4	5	6	7	8	9	10		
<b>Q1</b>	É simples visualizar os marcos no mapa.										4.5	0.50
<b>Q2</b>	Ícones usados para os marcos estão adequados à aplicação.										4.5	0.50
<b>Q3</b>	Os botões estão no local adequado do ecrã.										3.2	0.60
<b>Q4</b>	O tipo de letra usado na aplicação é adequado.										4.1	0.54
<b>Q5</b>	É útil obter a fotografia de uma câmara de trânsito (em tempo-real).										4.5	0.50
<b>Q6</b>	O botão de ajuda é importante.										4.3	0.46
<b>Q7</b>	É importante a sincronização com o Facebook.										4.5	0.50
<b>Q8</b>	A opção de aprovo/reprovo um marco é importante.										4.6	0.49
<b>Q9</b>	É simples adicionar um marco.										4.8	0.40
<b>Q10</b>	Encontrei rapidamente o botão de ajuda.										4.3	0.46
<b>Q11</b>	É simples consultar os marcos existentes.										4.2	0.40
<b>Q12</b>	Não preciso de explicações adicionais para aprender a usar aplicação.										4.5	0.50
<b>Q13</b>	O alerta de proximidade com um marco é dado no momento adequado.										4.6	0.49
<b>Q14</b>	A aplicação tem uma boa fluidez e tempos de resposta.										5	0.00
<b>Q15</b>	O dispositivo móvel não aquece quando a aplicação está aberta.										4.8	0.40
<b>Q16</b>	A aplicação deteta corretamente a minha posição no mapa.										4.3	0.46

Na versão da aplicação Estradas Seguras utilizada aquando do estudo de usabilidade os botões estavam dispostos de acordo com a Figura 26. Por defeito a aplicação mostrava o mapa de Portugal e os radares existentes (imagem à esquerda). Se o utilizador quiser visualizar as opções para adicionar novos marcos, sincronizar, etc. terá de clicar no botão menu do sistema Android. No menu da

aplicação (imagem à direita) o botão para adicionar um novo marco encontra-se em destaque. O motivo desta decisão está subjacente ao facto de os automobilistas poderem adicionar o marco com o mínimo de interação possível, ou seja, simplificar esta operação. Contudo, de acordo com o estudo de usabilidade existe um elevado grau de insatisfação com a localização dos botões. Na avaliação desta questão chegou-se à conclusão que a necessidade de clicar duas vezes para adicionar um marco era demasiado elevado. Por exemplo, o automobilista encontra-se em movimento e pretende rapidamente adicionar o ponto no mapa e o tempo perdido com mais do que um clique é relevante. Por isso, foi desenvolvida uma nova interface. Na Figura 27 é apresentada a nova interface, e é possível observar no canto superior esquerdo a existência de novos botões. Estes botões visam ser acessos rápidos às opções existentes no menu da aplicação (Figura 26 – imagem à direita). Na Tabela XVIII é apresentada uma descrição dos acessos rápidos adicionados à aplicação Estrada Segura.

**Tabela XVIII - Interface da aplicação - novas funcionalidades.**

<b>Ícone</b>	<b>Funcionalidade</b>
	Adicionar um novo marco.
	Atualizar os dados da aplicação.
	Radares fixos visíveis ou não no mapa.
	Acompanhar o deslocamento do dispositivo móvel (ligado/desligado).
	Botão de acesso às publicações do Facebook. As publicações são do distrito onde o automobilista se encontra.

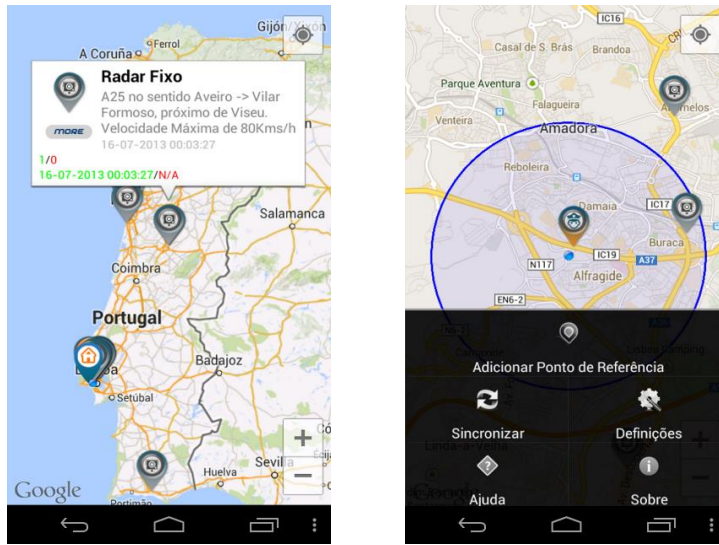


Figura 26 - Interface da aplicação no estudo de usabilidade.

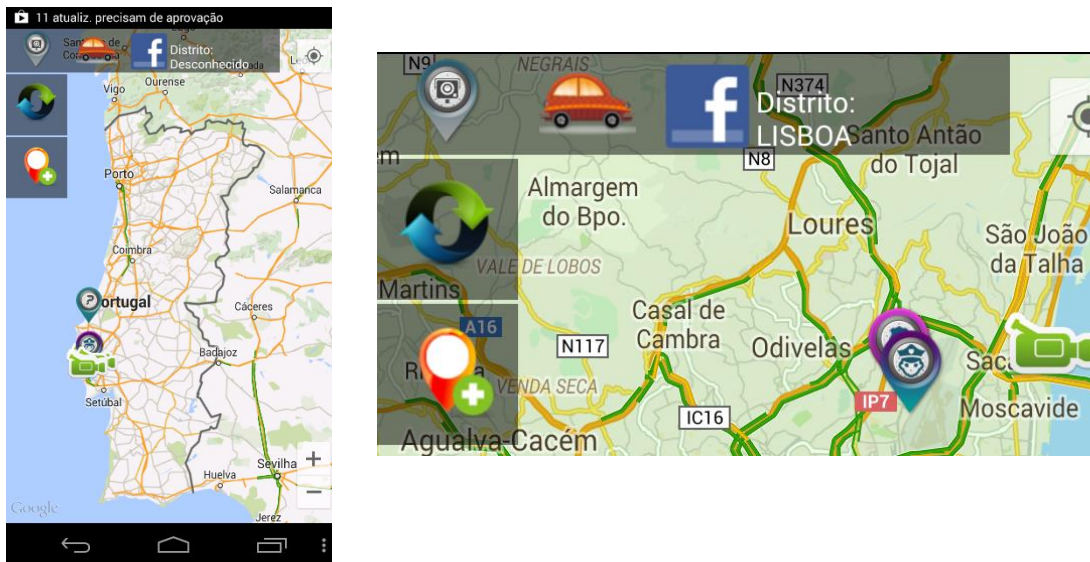


Figura 27 - Interface da aplicação após estudo de usabilidade

## 8. Conclusão

Nesta dissertação é apresentado um estudo na área aplicações móveis. Atualmente existe um mercado emergente onde tem vindo a aumentar a necessidade de melhorar a qualidade das aplicações disponíveis e, propor novas ideias de aplicações que utilizam tecnologias atuais que outrora não seria possível.

Apresentou-se uma implementação de uma aplicação móvel Estrada Segura que utiliza a triangulação das antenas e/ou a posição GPS para identificar a localização do automobilista. O objetivo da aplicação visa promover a divulgação de possíveis perigos na estrada através dos automobilistas. Portanto foi preciso desenvolver uma aplicação que apresenta-se as seguintes características:

- Uma interface simples e intuitiva;
- Eficientemente deteta-se a posição do automobilista;
- Calcula-se a proximidade a perigos no momento adequado;
- Salvaguarda-se a privacidade dos automobilistas;
- Integrasse tecnologias emergentes, como redes sociais.

Estes pontos foram as principais características que foram tidos em consideração durante o desenvolvimento da aplicação. Posteriormente foi realizado um estudo de utilizador com pessoas de diferentes áreas (estudos, idade, género e nível tecnológico) para avaliarem a aplicação em diferentes aspetos: interface, utilidade dos componentes disponíveis na aplicação, facilidade de aprendizagem e velocidade a realizar as diferentes tarefas. De forma geral os participantes demonstraram bastante agrado com a aplicação e este estudo possibilitou detetar melhorias a realizar no aspeto da interface. Finalmente foi feita uma análise sobre o ritmo de descarregamentos da aplicação na loja play da Google. O número de utilizadores tem vindo a crescer a um ritmo muito positivo.

Pode-se concluir que aplicações que apresentam uma interface apelativa e não necessitem de uma curva de aprendizagem muito elevada têm uma maior probabilidade de ter sucesso. Também se pode concluir que a ideia proposta apresenta-se como uma necessidade de mercado, pois não usufruiu de nenhuma campanha publicitária e tem apresentado um ritmo de crescimento muito elevado. E, finalmente que desta forma podemos prevenir possíveis agravamentos de perigos na estrada e/ou evitar cansaço por parte dos automobilistas no retorno a casa, pois podem escolher vias alternativas atempadamente.

## 8.1. Trabalho Futuro

No decorrer do desenvolvimento e avaliação da aplicação observou-se que utilizar um mapa alternativo ao Google maps (e.g. open-street-map) poderá aumentar o nível de satisfação de alguns utilizadores isto porque o mapa fica carregado no dispositivo móvel. Poderá existir utilizadores que têm um pacote de dados no dispositivo móvel baixo (limitação de MBs mensais) e, para estes utilizadores, descarregar os mapas do Google representa um elevado número de fluxo de dados. Os mapas da Google em comparação com as comunicações necessárias para obter os marcos é muito mais elevado, os marcos apenas têm a posição e foi tido o cuidado de reduzir o número de pacotes utilizado para as respetivas comunicações com o dispositivo móvel e servidor sejam eficientes.

Em trabalho futuro seria interessante realizar novas reuniões com o Diretor das Estradas de Portugal. A integração das camaras em tempo-real das Estradas de Portugal seria deveras uma mais-valia para a aplicação Estrada Segura. E para as Estradas de Portugal também poderiam usufruir de informação em tempo-real sobre acidentes, vias obstruídas etc.

Por fim, seria interessante incorporar um GPS na aplicação Estrada Segura. As rotas poderiam ter em consideração os perigos na via e calcular percursos alternativos tendo esta informação em consideração.



## Referências Bibliográficas

- Aapo, H. (2008). *Active Objects in Symbian OS*. MS Thesis University of Tampere. Retrieved from [http://www.cs.uta.fi/research/theses/masters/Haapanen\\_Aapo.pdf](http://www.cs.uta.fi/research/theses/masters/Haapanen_Aapo.pdf)
- Abowd, G. D. (1999). Software Engineering Issues for Ubiquitous Computing. In *Proceedings of the 21st International Conference on Software Engineering* (pp. 75–84). New York, NY, USA: ACM. doi:10.1145/302405.302454
- Alencar, P., & Cowan, D. (2011). *Handbook of Research on Mobile Software Engineering: Design Implementation and Emergent Applications (2 Volumes)* (1st ed.). Engineering Science Reference.
- Alshehri, F., & Freeman, M. (2012). Methods for usability evaluations of mobile devices. *Science and Technology Studies Commons*.
- Alur, D., Malsk, D., & Crupi, J. (2001). *Core J2EE Patterns: Best Practices and Design Strategies*. Upper Saddle River, NJ, USA: Prentice Hall PTR.
- Böckle, G., van der Linden, F. J., & Pohl, K. (2005). *Software product line engineering: foundations, principles and techniques*. Springer Science & Business Media.
- Boehm, B., & Belz, F. (1990). Experiences with the Spiral Model As a Process Model Generator. In *Proceedings of the 5th International Software Process Workshop on Experience with Software Process Models* (pp. 43–45). Los Alamitos, CA, USA: IEEE Computer Society Press. Retrieved from <http://dl.acm.org/citation.cfm?id=317498.317694>
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-oriented Software Architecture: A System of Patterns*. New York, NY, USA: John Wiley & Sons, Inc.
- Butler, K. A., Jacob, R. J. K., & John, B. E. (1998). Human-computer Interaction: Introduction and Overview. In *CHI 98 Conference Summary on Human Factors in Computing Systems* (pp. 105–106). New York, NY, USA: ACM. doi:10.1145/286498.286556
- Chen, P. P.-S. (1976). The Entity-relationship Model -Toward a Unified View of Data. *ACM Trans. Database Syst.*, 1(1), 9–36. doi:10.1145/320434.320440
- Choi, J. H., & Lee, H.-J. (2012). Facets of Simplicity for the Smartphone Interface: A Structural Model. *Int. J. Hum.-Comput. Stud.*, 70(2), 129–142. doi:10.1016/j.ijhcs.2011.09.002
- Cupchik, G. (2001). Constructivist Realism: An Ontology That Encompasses Positivist and Constructivist Approaches to the Social Sciences. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 2.
- Dodig-Crnkovic, G. (2002). Scientific Methods in Computer Science. In *Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden*. Retrieved from <http://www.es.mdh.se/publications/375->
- Duh, H. B.-L., Tan, G. C. B., & Chen, V. H. (2006). Usability Evaluation for Mobile Device: A Comparison of Laboratory and Field Tests. In *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services* (pp. 181–186). New York, NY, USA: ACM. doi:10.1145/1152215.1152254
- Dumas, J. F., & Redish, J. C. (1993). *A Practical Guide to Usability Testing*. Westport, CT, USA: Greenwood Publishing Group Inc.

- Flick, U. (2005). *Métodos qualitativos na investigação científica* (2ª ed.). Monitor. Retrieved from <http://books.google.es/books?id=PSrcygAACAAJ>
- Franklin, M. (2012). *Understanding research: coping with the quantitative - qualitative divide*. Routledge.
- Franzago, M., Muccini, H., & Malavolta, I. (2014). Towards a Collaborative Framework for the Design and Development of Data-intensive Mobile Applications. In *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems* (pp. 58–61). New York, NY, USA: ACM. doi:10.1145/2593902.2593917
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Giguère, E. (2001). Mobile Data Management: Challenges of Wireless and Offline Data Access. In *Proceedings of the 17th International Conference on Data Engineering* (p. 227–). Washington, DC, USA: IEEE Computer Society. Retrieved from <http://dl.acm.org/citation.cfm?id=876881.879717>
- Goadrich, M. H., & Rogers, M. P. (2011). Smart Smartphone Development: IOS Versus Android. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education* (pp. 607–612). New York, NY, USA: ACM. doi:10.1145/1953163.1953330
- Helander, G., Landauer, T. K., & Prabhu, P. V. (1997). *Handbook of Human-Computer Interaction*. Elsevier Science. Retrieved from <http://books.google.es/books?id=WuQbERgXR10C>
- Humphrey, W. S. (1989). Modelling implications of the personal software process. In *Proceedings of the Fifth International Software Process Workshop {(ISPW} '89), October 1989, Kennebunkport, Maine, {USA}* (pp. 74–77).
- Hyun, E.-H., & Kim, S.-H. (1995). Real-time mobile data management using a minimal MMDB. In *Real-Time Computing Systems and Applications, 1995. Proceedings., Second International Workshop on* (pp. 125–129). doi:10.1109/RTCSA.1995.528761
- Johnson, R. (2004). *Expert one-on-one J2EE design and development*. John Wiley & Sons.
- Jordan, P. W. (1998). *An Introduction To Usability*. Taylor & Francis. Retrieved from [http://books.google.es/books?id=WlkcQ\\_ukKwC](http://books.google.es/books?id=WlkcQ_ukKwC)
- Kelle, U. (2005). Sociological explanations between micro and macro and the integration of qualitative and quantitative methods. *Historical Social Research*, 30(1), 95–117. Retrieved from <http://nbn-resolving.de/urn:nbn:de:0168-ssoar-50194>
- Kim, H., Ryu, M., & Ramachandran, U. (2012). What is a Good Buffer Cache Replacement Scheme for Mobile Flash Storage? *SIGMETRICS Perform. Eval. Rev.*, 40(1), 235–246.
- Kimber, J., Georgievski, M., & Sharda, N. (2005). Developing Usability Testing Systems and Procedures for Mobile Tourism Services. *HITA -Proceedings Hospitality Information Technology Association Conference*, 79–96.
- Kjeldskov, J., & Graham, C. (2003). A Review of Mobile HCI Research Methods. In L. Chittaro (Ed.), *Human-Computer Interaction with Mobile Devices and Services* (Vol. 2795, pp. 317–335). Springer Berlin Heidelberg. doi:10.1007/978-3-540-45233-1\_23

- Li, H., & Zhijian, L. (2010). The study and implementation of mobile GPS navigation system based on Google Maps. In *Computer and Information Application (ICCIA), 2010 International Conference on* (pp. 87–90). doi:10.1109/ICCIA.2010.6141544
- Manolopoulos, V., Papadimitratos, P., Tao, S., & Rusu, A. (2011). Securing smartphone based ITS. In *ITS Telecommunications (ITST), 2011 11th International Conference on* (pp. 201–206). doi:10.1109/ITST.2011.6060053
- Manolopoulos, V., Tao, S., Rodriguez, S., Ismail, M., & Rusu, A. (2010). MobiTraS: A mobile application for a Smart Traffic System. In *NEWCAS Conference (NEWCAS), 2010 8th IEEE International* (pp. 365–368). doi:10.1109/NEWCAS.2010.5604010
- Matuscheck, J. P. (2011). *Finding Points Within a Distance of a Latitude/Longitude Using Bounding Coordinates*.
- McCormick, Z., & Schmidt, D. C. (2012). Data Synchronization Patterns in Mobile Application Design. In *Pattern Languages of Programs (PLoP) 2012 conference, October* (pp. 19–21).
- Meeker, M., Pitz, B., Fitzgerald, B., & Ji, R. (2005). Internet Trends. Retrieved from [http://www.web2con.com/presentations/web2con05/meeker\\_mary.pdf](http://www.web2con.com/presentations/web2con05/meeker_mary.pdf)
- Murphy, M. L. (2010). *Android Programming Tutorials* (3rd ed.). CommonsWare, LLC.
- Nilsson, E. G. (2009). Design Patterns for User Interface for Mobile Applications. *Adv. Eng. Softw.*, 40(12), 1318–1328. doi:10.1016/j.advengsoft.2009.01.017
- Nivala, A.-M. (2005). User-centred design in the development of a mobile map application. *Licentiate Thesis. Helsinki University of Technology, Department of Computer Science and Engineering*, 74p.
- Ohr, J., & Turau, V. (2012). Cross-Platform Development Tools for Smartphone Applications. *Computer*, 45(9), 72–79. doi:http://doi.ieeecomputersociety.org/10.1109/MC.2012.121
- Polatidis, N., & Georgiadis, C. (2014). Factors Influencing the Quality of the User Experience in Ubiquitous Recommender Systems. In N. Streitz & P. Markopoulos (Eds.), *Distributed, Ambient, and Pervasive Interactions SE - 35* (Vol. 8530, pp. 369–379). Springer International Publishing. doi:10.1007/978-3-319-07788-8\_35
- Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction Design* (1st ed.). New York, NY, USA: John Wiley & Sons, Inc.
- Pyke, T. N., & Blanc, R. P. (1973). Computer networking technology - A state of the art review. *Computer*, 6(8), 13–20. doi:10.1109/MC.1973.6541670
- Ribeiro, A., & da Silva, A. R. (2012). Survey on Cross-Platforms and Languages for Mobile Apps. In *Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the* (pp. 255–260). doi:10.1109/QUATIC.2012.56
- Sherman, M. (2014). An Introduction to Mobile Payments: Market Drivers, Applications, and Inhibitors. In *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems* (pp. 71–74). New York, NY, USA: ACM. doi:10.1145/2593902.2593921
- Standardization, I. O. F. (1998). *ISO 9241-11: 1998: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 11: Guidance on Usability*. International Organization for Standardization. Retrieved from <http://books.google.es/books?id=TzXYZwEACAAJ>

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the Qualitative-quantitative Divide: Guidelines for Conducting Mixed Methods Research in Information Systems. *MIS Q.*, 37(1), 21–54. Retrieved from <http://dl.acm.org/citation.cfm?id=2481690.2481693>

Wang, Y., & Ma, J. (2014). *Mobile Social Networking and Computing: A Multidisciplinary Integrated Perspective* (1st ed.). Boston, MA, USA: Auerbach Publications.

Wenderlich, R., Baranski, S., Burkepile, A., Gundersen, J., Hollemans, M., Marsetti, F. L., ... Todorov, M. (2013). *iOS 5 by Tutorials* (2nd ed.). USA: CreateSpace Independent Publishing Platform.

Zaharakis, I., & Komninos, A. (2012). Ubiquitous computing multidisciplinary endeavour. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 10(3), 1850–1852. doi:10.1109/TLA.2012.6222593