

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

**Facebike:
Agregação de Informação de Utilização
de Velocípedes via Rede Social de
Ciclistas**

Daniel Ferreira de Almeida

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Rosaldo José Fernandes Rossetti (PhD)

Orientador: Luís António Diniz Fernandes de Morais Sarmiento (PhD)

25 de Julho de 2011

**Facebike:
Agregação de Informação de Utilização de Velocípedes via
Rede Social de Ciclistas**

Daniel Ferreira de Almeida

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente: Maria Eduarda Silva Mendes Rodrigues (Doutora)

Vogal Externo: Fernando Augusto e Silva Mouta (Doutor)

Orientador: Rosaldo José Fernandes Rossetti (Doutor)

25 de Julho de 2011

Resumo

Nos dias que correm, a população mundial sente cada vez mais a necessidade de utilizar a bicicleta como meio de transporte. Este facto está ligado a razões económicas, à necessidade de evitar filas de trânsito e de praticar exercício físico. A população de velocipedes enfrenta no seu dia-a-dia alguns entraves à circulação em ambientes urbanos ou rurais. Alguns destes problemas podem ser solucionados pelas entidades responsáveis pela mobilidade numa determinada região. No entanto, nem sempre estas possuem a informação necessária para poder solucionar os problemas enfrentados pelos ciclistas.

Neste documento, é abordada uma possível solução para a problemática da deslocação dos ciclistas. A solução passa pela construção de um sistema constituído por duas aplicações *web*, uma que terá a função de recolha dos dados e a segunda para a análise destes. A aplicação de recolha de dados tem o propósito de agregar a informação sobre as deslocações efectuadas pelos ciclistas através de uma rede social. O Facebook foi a rede social escolhida para este efeito. A possibilidade que esta rede social oferece para a construção de aplicações aliada ao crescente número de utilizadores pode ser um factor determinante para uma boa recolha de dados.

Este sistema tem dois objectivos principais. Um é a construção de funcionalidades que sejam de elevado interesse para os ciclistas, melhorando as suas deslocações diárias. O outro é a extracção de conhecimento dos dados recolhidos através de técnicas de *Data Mining* visual.

Durante a implementação do sistema recorreu-se à manipulação de dados espaço-temporais. Para armazenar os dados foi utilizado o gestor de base de dados *PostgreSQL* com a extensão para dados geográficos *PostGIS*. Recorreu-se também à *API* do *Google Maps* para a sua respectiva visualização.

O protótipo do sistema implementado permite a agregação e análise dos dados das movimentações dos ciclistas, cumprindo os principais objectivos propostos. Para que este possa entrar em funcionamento junto dos seus utilizadores finais é necessário ainda implementar um conjunto de funcionalidades e executar alguns testes funcionais e de usabilidade.

Agradecimentos

Um agradecimento especial à minha família pelo apoio prestado durante esta fase da minha vida. Aos meus amigos que sempre estiveram presentes. A toda a comunidade FEUP que contribuiu para que conseguisse alcançar esta etapa da minha formação académica. Finalmente, a todos os músicos cuja arte sempre me acompanhou no desenvolvimento desta dissertação.

Daniel Ferreira de Almeida

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Definição do problema	1
1.2.1	Agregação dos dados e extracção de conhecimento	2
1.2.2	Redes sociais <i>online</i> : uma breve resenha histórica	2
1.3	Objectivos	3
1.4	Metodologia de desenvolvimento do sistema	8
1.4.1	Visualização e Data Mining	8
1.4.2	Recursos	9
1.5	Contribuições esperadas	9
1.6	Estrutura do documento	9
2	Enquadramento teórico	11
2.1	Dados espaço-temporais	11
2.1.1	KML	12
2.1.2	GPX	13
2.2	Algoritmos para encontrar o caminho mais curto	14
2.2.1	Algoritmo de Dijkstra	14
2.2.2	Algoritmo A*	15
2.3	Fórmula de Haversine	16
2.4	<i>Graph API</i>	17
2.5	<i>Google Maps Javascript API V3</i>	17
3	Trabalho relacionado	21
3.1	Comparação entre aplicações <i>web</i>	21
3.2	Aplicações para visualização de dados e extracção de conhecimento	23
3.2.1	<i>Safe Road Maps</i>	23
3.2.2	<i>hotpads.com</i>	23
3.2.3	<i>UK Heat Map</i>	24
3.2.4	Aplicação na visualização e análise de dados extraídos	25
4	Descrição do sistema	27
4.1	Funcionalidades propostas para o sistema	27
4.1.1	Aplicação de recolha de dados (FacebikeDC)	27
4.1.2	Aplicação de visualização e análise de dados (FacebikeDM)	30
4.2	Arquitectura do sistema	31
4.2.1	Camada de visualização	31

CONTEÚDO

4.2.2	Camada lógica	32
4.2.3	Camada de dados	33
4.2.4	Tecnologias	33
4.3	Modelo de dados	33
4.3.1	Classes	34
4.3.2	Relações	34
4.3.3	Modelo relacional de dados	34
4.4	Implementação	36
4.4.1	Integração com o Facebook	36
4.4.2	Criação de rotas	37
4.4.3	Visualização de rotas através do <i>Google Maps</i>	37
4.4.4	Cálculo de distâncias e velocidades	38
4.4.5	Mapa de declives	39
4.4.6	Análise de dados (FacebikeDM)	40
4.4.7	Outras funcionalidades implementadas	41
5	Análise do sistema	43
5.1	Validação das funcionalidades do sistema	43
5.2	Ficheiros de teste	43
5.3	Visualização de uma rota	43
5.4	Cálculo de distância e velocidade	44
5.5	Gráfico de perfil da rota	46
5.6	Mapa de declives	47
5.7	Aplicação de visualização de dados (FacebikeDM)	48
6	Conclusões e trabalho futuro	51
6.1	Satisfação dos objectivos	51
6.2	Dificuldades	52
6.3	Trabalho futuro	52
	Referências	55
A	Modelo de dados	57
A.1	Tabelas da base de dados	57
B	Esquema do modelo relacional de dados	61
C	Script para criação da base de dados	63

Lista de Figuras

1.1	Vista de criação/edição de rota do <i>website</i> www.bikely.com	5
1.2	Vista de uma rota do <i>website</i> www.cycle-route.com	5
1.3	Vista de um mapa de rotas do <i>website</i> www.nycbikemaps.com	5
1.4	Vista de um mapa de rotas do <i>website</i> www.labikepaths.com	6
1.5	Vista de um mapa de rotas do <i>website</i> www.sharemyroutes.com	6
1.6	Vista de um mapa de rotas do <i>website</i> www.everytrail.com	7
1.7	Vista de um mapa de rotas do <i>website</i> www.mapmyride.com	7
2.1	Árvore das classes da linguagem <i>KML</i> [13]	12
2.2	Formula de Haversine	17
2.3	Exemplo do uso das classes <i>Map</i> e <i>Polyline</i>	18
2.4	Exemplo do uso da classe <i>Mark</i> e <i>InfoWindow</i> [21]	18
2.5	Exemplo do uso da classe <i>Polygon</i> no Triângulo das Bermudas [21]	19
3.1	Visualização de um <i>heat map</i> na aplicação <i>Safe Road Maps</i>	23
3.2	Visualização da informação de um acidente na aplicação <i>Safe Road Maps</i>	24
3.3	Visualização da informação de uma zona de imóveis através da aplicação <i>UK Heat Map</i>	24
3.4	Visualização da informação de energia gasta da zona de Londres através da aplicação <i>UK Heat Map</i>	25
4.1	Arquitectura do sistema <i>Facebike</i>	32
4.2	Modelo de classes do sistema	34
4.3	Tabelas e relações da base de dados do sistema	35
4.4	Página de autorização de acesso	37
4.5	Visualização de uma rota através da aplicação <i>FacebikeDC</i>	38
4.6	Exemplo de um ficheiro <i>KML</i> visto a partir do <i>Google Earth</i>	40
5.1	Vista da rota do ficheiro <i>2007_06_24_RioLeca.gpx</i> através do <i>Google Earth</i>	44
5.2	Vista da rota do ficheiro <i>2007_06_24_RioLeca.gpx</i> através da aplicação <i>FacebikeDC</i>	44
5.3	Vista da rota do ficheiro <i>2007_07_21_EstVelocipedia40.gpx</i> através do <i>Google Earth</i>	45
5.4	Vista da rota do ficheiro <i>2007_07_21_EstVelocipedia40.gpx</i> através da aplicação <i>FacebikeDC</i>	45
5.5	Vista da rota do ficheiro <i>2007_06_24_RioLeca.gpx</i> através do <i>FacebikeDC</i>	46
5.6	Vista da rota do ficheiro <i>2007_06_24_RioLeca.gpx</i> através da aplicação <i>Gps Visualizer</i>	46

LISTA DE FIGURAS

5.7	Vista da rota do ficheiro 2007_07_21_EstVelocipedia40.gpx através do <i>Maplorer</i>	47
5.8	Comparação do mapa gerado com indicação através de cores do declive com <i>Google Maps</i>	47
5.9	Comparação entre o gráfico de perfil gerado através da API do Google Maps e do gerado a partir do ficheiro GPX na aplicação <i>Maplorer</i>	48
5.10	Resultado visual da aplicação FacebikeDM com aumento de resolução das rotas	49
5.11	Resultado visual da aplicação FacebikeDM com aumento de resolução das rotas	49
5.12	Resultado visual da aplicação FacebikeDM sem aumento de resolução das rotas	49
5.13	Resultado visual da aplicação FacebikeDM sem aumento de resolução das rotas	50
B.1	Tabelas e relações da base de dados do sistema	62

Lista de Tabelas

3.1	Tabela de comparação de aplicações.	22
4.1	Quadro resumo das funcionalidades do sistema Facebike	31
5.1	Tabela comparativa de velocidades e distâncias	46
A.1	Tabela fbuser	57
A.2	Tabela track_gps	57
A.3	Tabela point_gps	58
A.4	Tabela route	58
A.5	Tabela point	58
A.6	Tabela connection	59
A.7	Tabela route_comment	59
A.8	Tabela favourite	59
A.9	Tabela fbevent	59
A.10	Tabela profile	60
A.11	Tabela fbgroup	60

LISTA DE TABELAS

Definições, Acrónimos e Abreviaturas

Log	Registo de actividades no sistema
GPS	<i>Global Positioning System</i>
Javascript	Linguagem de <i>script</i> com orientação a objectos para desenvolvimento de aplicações <i>web</i>
Web Service	Serviço que permite a comunicação entre aplicações desenvolvidas em diferentes plataformas
Open Source	<i>Software</i> desenvolvido sem fins comerciais
SDK	<i>Software Development Kit</i> Facilita a integração dos programadores com um determinado <i>Software</i>
PHP	<i>Hypertext Preprocessor</i> Linguagem de <i>script</i> para desenvolvimento <i>web</i> de páginas com conteúdo dinâmico
Heat maps	Representação gráfica de uma determinada variável através de um código de cores
Microsoft .NET	Plataforma única para desenvolvimento e execução de sistemas e aplicações

ABREVIATURAS E SÍMBOLOS

Capítulo 1

Introdução

1.1 Contexto

Actualmente, a população de ciclistas tem estado em crescimento um pouco por todo o mundo. Segundo alguns estudos, grandes centros urbanos como Barcelona, Berlin, Bucharest, Roma, Gotemburgo [1], Southampton, Toronto [2], Portland [3] e São Francisco [4] assistiram ao crescimento do número de ciclistas nos últimos anos. As razões apresentadas para tal facto prendem-se maioritariamente à subida dos preços dos combustíveis [5], evitar as filas de trânsito e à prática de exercício físico [4].

1.2 Definição do problema

A bicicleta é um meio de transporte utilizado para diferentes finalidades. Este veículo é usado por muitas pessoas como uma forma de descontração aliada à prática de exercício físico. Pode também ser usado como meio de transporte para permitir a deslocação de um determinado ponto para outro. Para fins profissionais, a participação em competições internacionais é um dos principais exemplos. Sendo assim, de acordo com a finalidade atribuída à bicicleta, os seus utilizadores podem-se classificar em três grupos: lazer/ desporto, meio de transporte e profissional.

Consequentemente, cada grupo de utilizadores de bicicleta tem as suas necessidades particulares. É de prever que quem se desloca simplesmente por lazer tenha algum interesse em obter informação sobre a existência de algum ponto turístico no percurso que esta a efectuar. Poderá também ter interesse em saber se algum grupo de ciclistas pretende percorrer uma determinada rota em conjunto. Os ciclistas que utilizam a bicicleta como um meio de transporte, necessita de tomar decisões sobre qual o percurso que deverão efectuar para mais eficazmente se deslocarem. Os ciclistas de cariz profissional poderão

estar mais interessados em métricas como a velocidade média, a distância percorrida e o declive de uma determinada rota.

Diariamente, a comunidade de ciclistas enfrenta alguns desafios à deslocação em ambientes urbanos ou rurais. Alguns exemplos são o mau estado da via onde circulam, a existência ou não de vias destinadas a ciclistas, a tecnicidade do percurso que estão a percorrer (inclinação, tipo de piso, etc.), a quantidade de tráfego automóvel, a distância, o tempo do percurso e a segurança pessoal[6] [4]. Alguns destes problemas poderão obter solução junto das entidades públicas responsáveis pela mobilidade e infraestruturas de uma determinada região. Estes terão também a necessidade de identificar estes pontos para poder tomar medidas. A identificação destes pontos críticos é habitualmente feita através da recolha de dados junto da população ou por observação das movimentações efectuadas pelos ciclistas. Esta tarefa seria bastante dispendiosa e os seus resultados pouco abrangentes, uma vez que é impraticável numa grande cidade uma observação fiável de todas as rotas efectuadas. Para solucionar este problema, nesta dissertação pretende-se demonstrar que através de uma rede social *online* será possível agregar a informação necessária para posteriormente ser analisada, constituindo conhecimento relevante para as partes interessadas.

1.2.1 Agregação dos dados e extracção de conhecimento

Os dados a serem recolhidos são respeitantes a comportamentos e movimentações efectuados pelos ciclistas. Sendo assim, será necessário obter uma sequência de pontos a que corresponde uma determinada latitude e longitude para assim identificar o caminho que cada ciclista efectuou. Agregando também a estes pontos a informação sobre a data e hora a que foram atravessados, a recolha deste tipo de informação será possível através de um dispositivo GPS e respectivos *logs* das rotas percorridas.

Como dito anteriormente, será através de uma rede social *online* que os dados serão recolhidos, sendo depois transformados em informação e posteriormente em conhecimento para as partes envolvidas. Os dados por si só não têm significado, sendo que quando estes se encontram inseridos num contexto e depois de devidamente filtrados poderão ser constituídos como informação, à qual poderá ser atribuído um significado, constituindo assim conhecimento [7].

1.2.2 Redes sociais *online*: uma breve resenha histórica

Segundo um estudo sobre redes sociais *online* [8], a primeira rede social *online* foi criada em 1987 quando os colaboradores da GreenNet¹ começaram a trabalhar em conjunto com o *Institute for Global Communications (IGC)*. Partilhando material electrónico, esta

¹A GreenNet é uma iniciativa sem fins lucrativos sediada em Inglaterra que trabalha em prol da paz, protecção do meio ambiente e direitos do homem

rede colaborativa rapidamente se estendeu a outras organizações e países. Mais tarde, em 1994, surge o Geocities, uma plataforma que permitia ao utilizador a criação de um *website* respeitante a uma determinada categoria para partilha na *web*. Em 1995, pela mão de estudantes da Universidade de Cornell surge o site *theglobe.com*, que oferecia aos seus utilizadores a possibilidade de interagirem com outros com interesses em comum, possibilitando também a partilha de conteúdos pessoais. No ano de 1997, aparece o *AOL Instant Messenger*, possibilitando aos utilizadores a troca de mensagens em tempo real. Esta inovação fez com que este tipo de serviço fosse bastante popularizado. A rede social *sixdegrees.com* é lançada em 1997, permitindo a criação do perfil de utilizador, bem como a listagem de amigos. Em 2002, surge o Friendster, que um ano mais tarde foi clonado por uma empresa de marketing *online*, dando assim origem às primeiras versões do Myspace.

A rede social Facebook foi lançada em 2004, por Mark Zuckerberg, na Universidade de Harvard. Originalmente, o Facebook surgiu como um veículo de ligação entre estudantes americanos. Hoje em dia é a rede social líder em termos de utilizadores registados e activos. Finalmente, em 2006 nasce a rede social Twitter, caracterizada pelo seu serviço de *microblogging* [9].

As redes sociais são nos dias que correm um fenómeno global. Comunidades como o Myspace ou Facebook albergam milhões de utilizadores². Através destas aplicações web os utilizadores são capazes de partilhar as suas informações pessoais e interesses de inúmeras formas, dando-se a conhecer a comunidades de pessoas com um número bastante considerável; facto que antes do *boom* da Internet era bastante improvável. Com esta partilha de informações os utilizadores vão estabelecendo ligações, não só com as pessoas que já fazem parte do seu círculo de amigos mas também com outras que partilhem os mesmos interesses. A troca de informação está bastante facilitada, o que permite a criação de redes colaborativas em torno de um grupo de pessoas que tenham interesses em comum.

1.3 Objectivos

O sistema Facebike é constituído por duas aplicações. Uma integrada na rede social Facebook, que tem como objectivo primordial a agregação de informação dos utilizadores de bicicleta. A outra será uma aplicação de análise dos dados recolhidos. Através destes dados pode-se inferir qual o comportamento em termos de movimentações demográficas dos ciclistas em ambiente urbano ou rural, tendo em vista um melhor planeamento de mobilidade destes. Os utilizadores finais serão preferencialmente pertencentes ao grupo de pessoas que tenham interesse pela prática de ciclismo, quer seja como forma de lazer, meio de transporte ou profissional. Através desta aplicação irá ser promovida a interacção

²<http://www.flowtown.com/blog/the-2010-social-networking-map?display=wide>

entre os ciclistas na partilha de informações relevantes para a prática deste desporto, com especial foco nas rotas que cada utilizador percorre nas suas viagens.

Para que o ciclista se sinta interessado em partilhar a informação sobre as rotas que percorre é necessário que o Facebike lhe forneça informação de valor para que este encontre vantagens em utilizar a aplicação. Sendo assim, a aplicação deverá permitir aos utilizadores a partilha das suas rotas com a possibilidade de estas também serem comentadas. A criação de uma rota poderá ser feita através de um log GPS ou através da própria aplicação. Em cada rota o ciclista poderá também assinalar e comentar os pontos/troços críticos do percurso. Sobre estas é também possível consultar alguns dados estatísticos relativos a velocidades, tempo do percurso, distância percorrida e altitudes nos diferentes pontos.

Aos utilizadores será também permitido criar eventos em que estes poderão convidar outros ciclistas com o objectivo de todos percorrerem um determinada percurso. A criação de grupos de ciclistas será também permitida na aplicação. Através da aplicação serão também dadas sugestões de amigos que possuam características em comum e sugestões de melhoramentos das suas rotas. Serão criados *rankings* de *performance* dos ciclistas que serão associados a vários perfis de utilizador, com o objectivo da aplicação se adaptar a cada tipo de ciclista.

Adicionalmente, o sistema deverá gerar, sobre todas as rotas dos ciclistas, mapas de dispersão que apresentam estatísticas de uma determinada zona tendo em conta velocidades médias, frequência de passagem, percentagem de inclinação e estado do piso.

Criado o valor para o utilizador, espera-se que haja uma boa adesão ao Facebike, maximizando assim a quantidade de informação disponível para análise. Para que tal aconteça, a usabilidade representa uma parte importante na aplicação. É dada bastante importância à facilidade com que o utilizador interage com a aplicação. A criação de mapas interactivos, para que os utilizadores possam fornecer informações com maior rigor, irá permitir a obtenção de resultados mais precisos da posterior análise.

Seguem-se alguns exemplos de aplicações *web* que cumprem alguns dos objectivos propostos para a recolha de dados junto dos ciclistas.

1.3.0.1 bikely.com

Este *website* (Figura 1.1) permite aos utilizadores previamente registados partilharem as suas rotas de interesse através da introdução manual numa interface que inclui o Google Maps ou através de um ficheiro do tipo GPX³. Os percursos partilhados podem ser marcados de acordo com a sua utilidade, sendo que estes podem também ser pesquisados por vários parâmetros como o seu tamanho ou localização.

³Formato XML para transferir dados GPS entre aplicações

Introdução

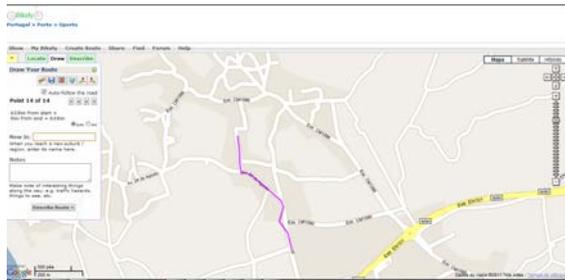


Figura 1.1: Vista de criação/edição de rota do *website* www.bikely.com

1.3.0.2 cycle-route.com



Figura 1.2: Vista de uma rota do *website* www.cycle-route.com

O cycle.route.com (Figura 1.2) permite a utilizadores não registados a introdução de rotas de maneira semelhante à usada pelo website bikely.com. É também possível acrescentar comentários a cada rota, bem como definir qual a qualidade e dificuldade da mesma.

1.3.0.3 nycbikemaps.com



Figura 1.3: Vista de um mapa de rotas do *website* www.nycbikemaps.com

Introdução

Neste *website* (Figura 1.3) é possível visualizar rotas de bicicleta de vários tipos na cidade de Nova Iorque. Não é possível um utilizador partilhar a sua própria rota, entretanto.

1.3.0.4 labikepaths.com

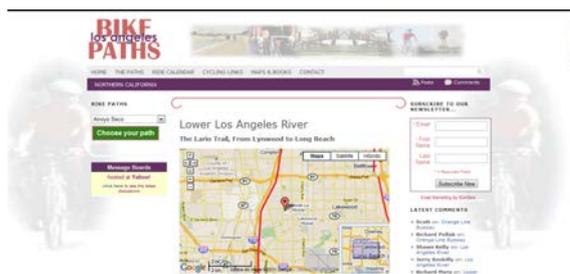


Figura 1.4: Vista de um mapa de rotas do *website* www.labikepaths.com

O conceito que envolve este *website* (Figura 1.4) é bastante semelhante ao do nycbike-maps.com, com a diferença de que em cada rota apresentada é possível ver uma descrição detalhada de todo o percurso, acompanhada de fotografias de alguns locais de interesse.

1.3.0.5 sharemyroutes.com



Figura 1.5: Vista de um mapa de rotas do *website* www.sharemyroutes.com

O sharemyroutes.com (Figura 1.5) é uma aplicação que permite ao utilizador registado partilhar a sua rota de qualquer tipo, através da introdução de um ficheiro. Podendo este estar em vários formatos. É possível criar uma rota através do editor de rotas. Esta aplicação fornece ainda bastante informação acerca da rota em questão, destacando-se um gráfico do perfil da rota que interage com a interface que pode ser proveniente do *Google Maps* ou *Bing Maps*. É ainda possível descarregar a rota em vários formatos (KML, GPS, CSV, etc.).

Introdução



Figura 1.6: Vista de um mapa de rotas do *website* www.everytrail.com

1.3.0.6 everytrail.com

Este *website* (Figura 1.6) de partilha de rotas permite aos utilizadores através de um método semelhante ao da aplicação sharemyroutes.com a partilha de rotas. O everytrail.com tem a particularidade de se poder carregar uma determinada rota através de um dispositivo GPS da marca *Garmin*. Adicionalmente, permite ao utilizador adicionar fotografias ou vídeos à sua rota.

1.3.0.7 mapmyride.com

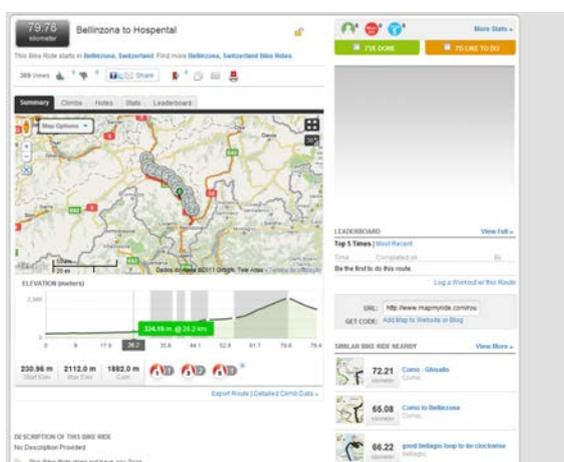


Figura 1.7: Vista de um mapa de rotas do *website* www.mapmyride.com

A aplicação *web* [mapmyride](http://mapmyride.com) apresenta uma enorme diversidade de funções. Permite a criação de rotas, visualização de estatísticas e sugestão de outras rotas. Promove a interação dos ciclistas através da criação de grupos e eventos. É possível também acompanhar uma rota de um ciclista em tempo real. Fornece também informações sobre alimentos e calorías gastas. Existem alguns serviços disponibilizados mediante o pagamento de uma

determinada quantia. Estes incluem, por exemplo planos de treinos e geração de relatórios destes. Aliado a estes factores, está a boa usabilidade que permite aos utilizadores construir e interagir com uma determinada rota de maneira bastantes intuitiva.

1.4 Metodologia de desenvolvimento do sistema

O sistema tem como principal objectivo a agregação e posterior análise de dados das movimentações de ciclistas, através de uma rede social. A rede social escolhida para este efeito foi o Facebook. A escolha recaiu nesta rede social porque, para além de ser a que mais utilizadores activos alberga, permite também o desenvolvimento de aplicações que interagem com os utilizadores do Facebook. A recolha de dados será feita com base nas rotas que os utilizadores introduzirem na aplicação e armazenada numa base de dados com características específicas para dados espaço-temporais. Posteriormente, a análise de padrões das informações recolhidas das deslocações de cada ciclista é feita através de técnicas de *data mining* aplicaas a dados espaço- temporais. Permite-se assim identificar, entre outras características, zonas críticas para a prática de ciclismo, concentração da população de ciclistas, optimização dos percursos de cada ciclista, frequência de passagem de velocípedes e respectivas velocidades, etc. A extracção de conhecimento dos dados recolhidos será apoiada fortemente numa componente visual que constituirá uma aplicação de visualização e análise de dados (FacebikeDM). Numa fase final, o sistema foi avaliado quanto à correcta execução das suas funcionalidades.

1.4.1 Visualização e Data Mining

O objectivo da visualização de dados de uma forma gráfica e interactiva é essencialmente a redução da sua complexidade. Desta forma o ser humano consegue mais facilmente procurar por estruturas, padrões e ligações entre os dados para extrair destes o conhecimento necessário. Segundo, [10] a visualização pode contribuir para o processo de Data Mining de três formas:

- permitir a visualização de resultados de algoritmos bastante complexos (visualização de dados provenientes de algoritmos de Data Minig);
- descobrir de padrões que não são identificáveis através de qualquer técnica, mas sim pelo ser humano, constituindo o Data Mining visual;
- através da combinação das anteriores.

1.4.1.1 Data Mining visual

Através deste tipo de Data Mining o ser humano irá formular uma hipótese sobre os padrões e relações existentes nos dados que, segundo [11] podem ser usados como análise

exploratória ou confirmativa. Na análise exploratória, o utilizador não tem qualquer conhecimento prévio sobre os dados que está a observar e tenta através destes construir uma hipótese que mais tarde será confirmada com recurso a outras técnicas. Por outro lado na análise confirmativa, o utilizador já possui algum conhecimento acerca dos dados que está a observar e assim pretende-se que este confirme ou não uma hipótese proposta anteriormente. O sistema Facebike é constituído por uma aplicação que aplica as técnicas de Data Mining visual na qual será aplicada tanto a análise exploratória como a confirmativa, dependendo dos objectivos do utilizador.

1.4.2 Recursos

Os recursos que foram utilizados nesta dissertação tiveram como função o auxílio à compreensão da temática em questão, o estudo das tecnologias necessárias ao desenvolvimento da aplicação e compreensão das necessidades dos utilizadores finais da aplicação. Foi feita a revisão de artigos e bibliografia relacionada com aspectos mais técnicos da implementação da aplicação. Foram também estudados alguns questionários realizados a comunidades de ciclistas em diversas cidades um pouco por todo o mundo. Para o desenvolvimento da aplicação foram também disponibilizados ficheiros de teste contendo dados sobre rotas efectuadas por um grupo de ciclistas.

1.5 Contribuições esperadas

Como já dito anteriormente, através da análise dos dados das movimentações geográficas é possível inferir qual o comportamento da comunidade de ciclistas, quer em ambiente urbano ou rural. Esta percepção poderá ter alguma importância para as autarquias. Estas poderão melhorar o planeamento de tráfego, o que fará com que as condições para o trânsito de velocípedes sejam melhoradas. Por outro lado, num cariz mais social, se a aplicação obtiver uma boa aceitação dos utilizadores do Facebook, poderá despertar o interesse de outras pessoas para a prática do ciclismo. A principal contribuição nesta dissertação esta no facto de aliar as redes sociais à agregação de dados e à melhoria da prática do ciclismo, quer através da interacção entre os ciclistas quer pela análise de dados das suas movimentações.

1.6 Estrutura do documento

Este documento é constituído por seis capítulos distintos. No primeiro capítulo são introduzidas as temáticas que serão abordadas na dissertação, definido o problema, objectivos e qual a metodologia a usar ao longo do estudo. No capítulo dois são apresentados

Introdução

alguns conceitos teóricos relevantes para esta dissertação. No terceiro capítulo são apresentados e analisados alguns trabalhos relacionados com esta dissertação e qual a possível contribuição destes para o desenvolvimento do sistema proposto. O quarto capítulo apresenta uma descrição do sistema desenvolvido. No capítulo cinco, o sistema desenvolvido foi analisado quanto à correcta execução das suas funcionalidades. Finalmente, no último capítulo são apresentadas as principais conclusões do trabalho desenvolvido e as indicações para o trabalho futuro.

Capítulo 2

Enquadramento teórico

O presente capítulo tem o objectivo de detalhar alguns dos conhecimentos teóricos necessários para a construção do sistema proposto. São apresentadas as características dos dados a utilizar, alguns algoritmos para pesquisa do caminho mais curto e as *API's* usadas na implementação do sistema.

2.1 Dados espaço-temporais

Os dados espaço-temporais pressupõem a junção de informação geográfica com informação temporal, ou seja, pode-se saber qual a posição exacta no tempo e no espaço de um determinado indivíduo ou objecto desde que este seja portador de um sistema que faça um registo deste tipo de dados. Estes dados são depois armazenados numa base de dados, tendo como objectivo de posteriormente serem analisados para que seja extraído conhecimento relevante. É necessário que a base de dados possua características que a diferenciam das bases de dados relacionais mais usuais. Nas bases de dados *standard* os dados vão sofrendo *updates* de uma forma mais explicita, ou seja, lidando com dados espaço-temporais seria necessário fazer actualizações aos dados constantemente. Para que isto seja evitado é guardada uma descrição das alterações dos dados em função do tempo [12].

Outras formas de representação de informações associadas a dados espaço-temporais, pode ser feita recorrendo a formatos baseados em *XML*. Para o desenvolvimento do Facebike os formatos utilizados foram o *Keyhole Markup Language (KML)* e o *GPS eXchange Format (GPX)*. A escolha destes formatos deveu-se principalmente, no caso do *KML* pela sua compatibilidade com o *Google Maps*, permitindo a sua visualização recorrendo à *API* da *Google*. O formato *GPX*, para além de ser bastante utilizado, é o formato no qual os registos das rotas que servirão de teste à aplicação se encontram.

2.1.1 KML

Este formato permite a descrição e anotação de dados geográficos. Pode ser usado em aplicações como o Google Earth, Google Maps ou Mobile. O Google Earth permite ao utilizador gerar este tipo de ficheiros através da colocação das anotações directamente num mapa. Na Figura 2.1 pode-se observar que tipo de geometrias (*object Geometry*) é possível descrever com esta linguagem. Na aplicação desenvolvida, as *tags* usadas foram: <Point>, <LineString> e <MultiGeometry>. A aplicação permite também que sejam definidos caminhos recorrendo à <LineString >, que alberga um conjunto de pontos identificados pela sua latitude, longitude e altitude. Através do objecto *SubStyle*, recorrendo à *tag* <LineStyle>, é possível definir, por exemplo, qual a cor e espessura da <LineString> que será marcada sobre o mapa.

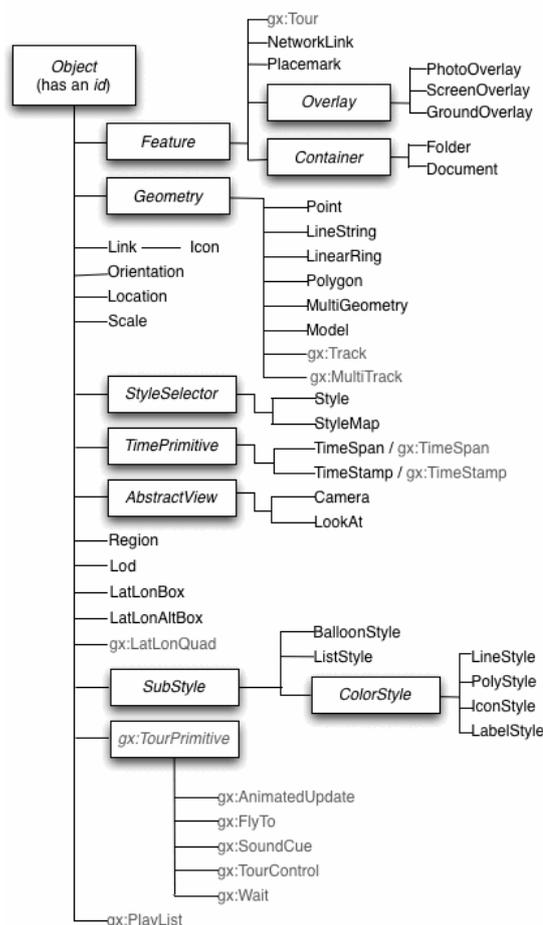


Figura 2.1: Árvore das classes da linguagem KML [13]

Em seguida encontra-se o código KML de um ficheiro em que é descrita uma linha constituída por dois pontos. Estes estão representados na *tag* <coordinates>, com os valores para longitude, latitude e altitude, respectivamente. Na *tag* <Style> está definida

a cor bem como a espessura da linha a apresentar no mapa. A listagem a seguir representa a estrutura de um ficheiro KML.

```
<?xml version="1.0" standalone="yes"?>
<kml xmlns="http://earth.google.com/kml/2.2">
  <Document>
    <Placemark>
      <MultiGeometry>
        <LineString>
          <coordinates>
            -8.6558961290000003,41.273081314999999,73.197630000000004
            -8.6556815530000009,41.273252976000002,68.871579999999994
          </coordinates>
        </LineString>
      </MultiGeometry>
      <Style>
        <LineStyle>
          <color>FF14F03C</color>
          <width>4</width>
        </LineStyle>
      </Style>
    </Placemark>
  </Document>
</kml>
```

2.1.2 GPX

O formato GPX é usado por dispositivos GPS para descrever dados relacionados com o registo das movimentações efectuadas sobre a Terra. Permite albergar a informação relacionada com *waypoints*, *routes* e *tracks*. *Waypoints* (<wpt>) são pontos que um utilizador de um aparelho GPS define como querendo que façam parte do seu percurso. Um conjunto de *waypoints* (<wpt>) ligados por vários troços constituem uma *route* (<rte>). Nesta linguagem, *tracks* é o conjunto dos pontos que foram atravessados pelo utilizador, quando este percorreu um determinado percurso [14]. A listagem a seguir exemplifica a estrutura de um ficheiro GPX.

```
...
<wpt lat="41.326032144" lon="-8.654779073">
  <ele>87.605100</ele>
  <time>2004-01-21T03:00:00Z</time>
```

Enquadramento teórico

```
<name>ES318</name>
<cmt>Estrada Secundária.</cmt>
<desc>Estrada Secundária.</desc>
<sym>Car</sym>
</wpt>
...
<trk>
<name>ACTIVE LOG</name>
<trkseg>
<trkpt lat="41.267545235" lon="-8.666217270">
<ele>65.987670</ele>
<time>2004-01-03T09:17:48Z</time>
</trkpt>
<trkpt lat="41.267566693" lon="-8.666260185">
<ele>66.468260</ele>
<time>2004-01-03T09:17:52Z</time>
</trkpt>
...
```

O código anteriormente representado é um excerto retirado de um dos ficheiros usados para testes à aplicação e pretende descrever uma pequena parte de um percurso. Este é constituído por um *waypoint* e dois *track points*, cuja informação em termos de tempo e espaço se encontra definida pelos atributos *lat* e *lon* e pelas *tags* `<ele>` e `<time>`, representando estas a latitude, longitude, altitude e data/hora em que o ponto foi marcado.

2.2 Algoritmos para encontrar o caminho mais curto

O problema de encontrar o caminho mais curto entre uma determinada origem e destino tem sido amplamente estudado ao longo dos anos. A sua aplicação na análise de redes de transporte é, nos dias que correm, amplamente reconhecida. Em seguida serão descritos alguns algoritmos que permitem o seu cálculo de forma eficaz.

2.2.1 Algoritmo de Dijkstra

Um dos algoritmos que vem sendo otimizado ao longo dos anos é o algoritmo de Dijkstra, que foi originalmente concebido por Edsger Dijkstra em 1956 e publicado em 1959 [15]. É um algoritmo de pesquisa em grafos com o objectivo de determinar o caminho mais curto entre um vértice de origem e um de destino. Um grafo direccional é representado por um conjunto de vértices conectados por arestas, em que a direcção em cada uma das arestas é representada pela ordem de ligação dos seus vértices. Cada aresta

contem também o custo. Este pode ser associado por exemplo à distância entre vértices. Neste algoritmo cada vértice irá ter um valor para o custo (distância) mínimo actual desde a origem.

2.2.1.1 Pseudo-código

O algoritmo apresentado a seguir foi retirado de [16].

1. Atribuir 0 ao valor do custo do vértice inicial (origem) e marca-lo como vértice actual. Todos os restantes terão como valor para o custo infinito e devem ser marcados como não visitados.
2. Para cada nó que ainda não foi visitado e que seja adjacente ao vértice actual, se o valor do nó actual mais o custo da aresta que liga ao nó adjacente for menor, então o custo total do vértice adjacente será alterado para esse valor, senão este permanece igual.
3. Marcar o vértice actual como visitado. Se ainda existirem vértices que ainda não foram visitados, então o novo vértice actual será o que tiver menor custo e volta-se ao passo 2. Se não existem mais nós por visitar, então o algoritmo acaba.

2.2.2 Algoritmo A*

O algoritmo A* surgiu na década de sessenta, numa tentativa de otimizar o algoritmo de Dijkstra. Peter E. Hart provou que a proposta feita por Nils Nilsson e aperfeiçoada por Bertram Raphael era de facto mais eficiente quando utilizada uma heurística consistente. Este algoritmo utiliza uma função de avaliação $f(n)$ para os nós a serem avaliados. Esta função é constituída pela soma de outras duas, $g(n)$ e $h(n)$, em que $g(n)$ é o custo da deslocação desde o ponto inicial até ao nó actual e $h(n)$ é a heurística que calcula qual o valor mínimo do custo até ao ponto de destino.

$$f(x) = g(x) + h(x)$$

A função heurística $h(n)$ é dita como consistente se para todos os nós n e cada sucessor s , $h(n) \leq C(n,s) + h(s)$, em que $C(n,s)$ é o custo mínimo de todos os caminhos entre n e s [17].

Neste algoritmo são também usadas duas listas de nós, nomeadamente uma de nós abertos e outra de nós fechados. A lista de nós abertos constitui os nós aos quais foi aplicada a função heurística mas que ainda não foram analisados. Esta lista está organizada numa fila de prioridades ordenada pelo valor mais promissor de cada nó. A lista de nós fechados alberga todos os nós que já foram analisados.

2.2.2.1 Pseudo-código para pesquisa em grafos

O algoritmo a seguir foi retirado de [18].

1. Criar um grafo de pesquisa G , somente com o nó inicial, n_0 . Colocar n_0 na lista de nós abertos.
2. Criar a lista de nós fechados, inicialmente vazia.
3. Se a lista de nós abertos estiver vazia, terminar com uma falha.
4. Seleccionar o primeiro nó na lista de abertos, removê-lo desta lista e colocá-lo na lista de fechados. Atribuir o nome n a este nó.
5. Se n for o nó de destino, então terminar com sucesso e obter a solução traçando um caminho de n para n_0 no grafo G .
6. Expandir o nó n , gerando o conjunto M dos seus sucessores. Definir os membros de M como sucessores de n no grafo G .
7. Definir um apontador para n a partir de cada um dos membros de M que ainda não estão em G (isto é, que não estão quer na lista de nós abertos ou na lista de nós fechados). Adicionar estes membros de M à lista de abertos. Para cada membro m de M que já estiver na lista de nós fechados ou abertos, redireccionar o seu apontador para n se o melhor caminho para m encontrado até agora passar por n . Para cada membro de M presente na lista de nós fechados, redireccionar o seu apontador para os seus descendentes em G , para que estes apontem para o melhor caminho encontrado para os descendentes.
8. Reordenar a lista de abertos por ordem crescente dos valores de $f(n)$.
9. Voltar ao passo 3.

2.3 Fórmula de Haversine

A fórmula de Haversine (Figura 2.2) permite calcular a distância entre dois pontos numa superfície esférica. Considerando que a terra pode ser aproximada a uma superfície esférica é possível, recorrendo a esta formula, calcular a distância entre dois quaisquer pontos na sua superfície, através dos valores para a latitude e longitude destes. Na fórmula da figura 2.2 R representa o raio da Terra, enquanto lat e $long$ são a latitude e longitude de cada um dos pontos [19].

$$a = \sin^2(\Delta lat/2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2(\Delta long/2)$$

$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = R \cdot c$$

Figura 2.2: Formula de Haversine

2.4 Graph API

A *Graph API* é um serviço disponibilizado pelo Facebook para aceder a informações presentes no *grafo social* desta rede social. Os pedidos de acesso são feitos através de um *Web Service* cuja resposta é dada em formato *JSON*. Por exemplo, para aceder a informações públicas de um utilizador com o *username* btaylor basta fazer um pedido através do link <https://graph.facebook.com/btaylor>. Os objectos contidos no "grafo social" do Facebook contêm um identificador pelo qual podem também ser acedidos. Por exemplo para o identificador 19292868552 basta aceder ao link <https://graph.facebook.com/19292868552>. Existem doze tipos de objectos: *Users, Pages, Events, Groups, Applications, Status messages, Photos, Photo albums, Profile pictures, Videos, Notes* e *Checkins*. Cada um destes objectos encontra-se conectado por uma determinada relação que pode ser de vários tipos, dependendo também do tipo de objecto: *Friends, News feed, Profile feed (Wall), Likes, Movies, Music, Books, Notes, Permissions, Photo Tags, Photo Albums, Video Tags, Video Uploads, Events, Groups* e *Checkins*.

Para que seja possível aceder à informação que um utilizador não tornou pública é necessário que *access token* seja incorporado no pedido à *Graph API*. Por exemplo, para que seja possível visualizar a informação do utilizador btaylor é necessário adicionar ao pedido um campo contendo essa chave de acesso. Neste caso o link seria https://graph.facebook.com/btaylor?access_token=AcessToken, em que *AcessToken* seria substituído pela respectiva chave de acesso. Para facilitar o acesso a esta API o Facebook disponibiliza vários *Software Development Kit (SDK)* para aplicações *web* e *móveis*: *JavaScript SDK, iOS SDK (iPhone & iPad), Android SDK* e *PHP SDK* [20].

2.5 Google Maps Javascript API V3

Esta API torna possível a incorporação do *Google Maps* numa aplicação *web*. Proporciona uma série de funcionalidades que permitem manipular mapas e acrescentar conteúdo através de vários serviços. O primeiro passo para se poder integrar a interface *Google Maps* é a criação de um objecto do tipo *google.maps.Map* que irá criar uma interface semelhante à que se pode obter através do website <http://maps.google.com/>. Nesta interface é possível desenhar linhas através da classe *google.maps.Polyline*. Esta

Enquadramento teórico

permite a definição de conjuntos de pontos que serão unidos formando um possível caminho, como se pode observar pela Figura 2.3 na linha a preto [13].

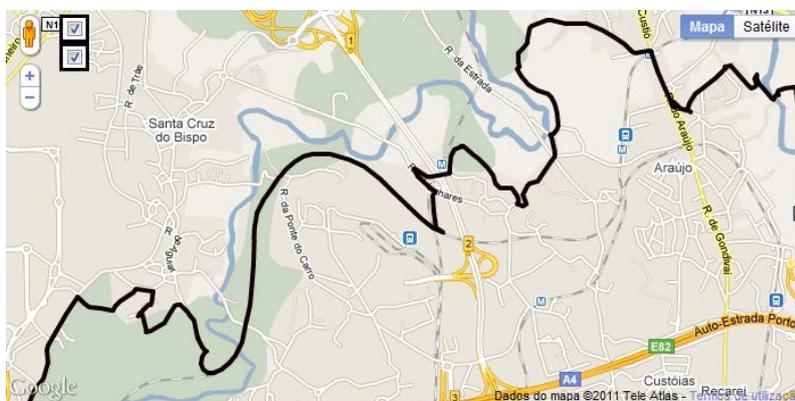


Figura 2.3: Exemplo do uso das classes *Map* e *Polyline*

A marcação de localizações no mapa pode ser feita recorrendo à classe *google.maps.Marker*. Através desta é possível desenhar um determinado *icon*, definido pelo utilizador, na localização pretendida, fornecendo para o efeito as coordenadas (latitude e longitude) desse ponto. A cada marca pode ser associada uma janela de informação através da classe *google.maps.InfoWindow*. O conteúdo desta janela pode ser alterado com qualquer informação textual. Permite, por exemplo, a incorporação de código *HTML* que pode conter elementos para a interação com o utilizador.

Cada objecto enunciado anteriormente pode conter um *event listener* através da função *google.maps.event.addListener*, permitindo que, por exemplo ao clicar com o rato num desses objectos, seja executada uma determinada acção. O exemplo na Figura 2.4 contém uma marca à qual foi associada um *event listener*. Ao clicar nesta apareça uma janela de informação com o texto sobre a localização.



Figura 2.4: Exemplo do uso da classe *Mark* e *InfoWindow* [21]

Enquadramento teórico

Através desta *API* é também possível desenhar polígonos (Figura 2.5) num mapa, recorrendo à classe *google.maps.Polygon*. Esta permite que seja definido um qualquer polígono somente através dos seus pontos. É possível também que características como a cor ou o seu contorno sejam editadas.

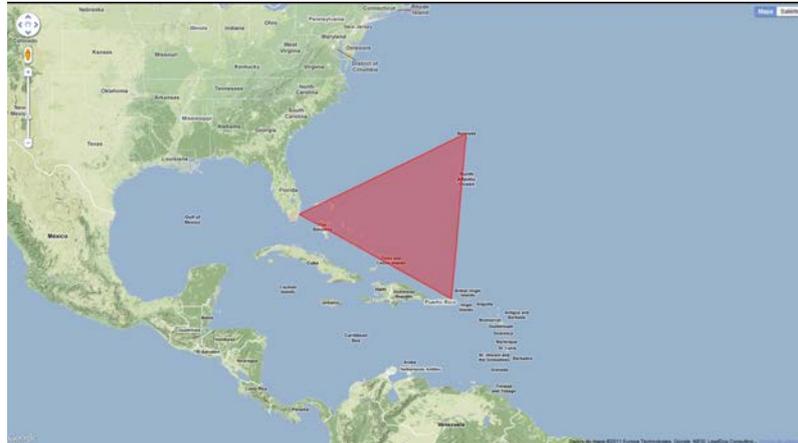


Figura 2.5: Exemplo do uso da classe *Polygon* no Triângulo das Bermudas [21]

Enquadramento teórico

Capítulo 3

Trabalho relacionado

Neste capítulo é feita uma comparação entre as aplicações *web* que servem os mesmos propósitos da aplicação Facebike. São também identificadas as vantagens que esta tem em relação às demais. Finalmente, são analisadas algumas aplicações constituídas por *heat maps* interactivos, com o intuito de extrair conhecimento de como interagir com os utilizadores finais da aplicação de visualização e análise de dados, FacebikeDM.

3.1 Comparação entre aplicações *web*

Na Tabela 3.1 é identificada a presença ou não de uma determinada característica ou funcionalidade nas aplicações descritas no Capítulo 1 secção ???. As característica ou funcionalidades presentes na comparação são as seguintes:

1. **Criação/Edição de rotas:** permite ao utilizador criar e editar os pontos que constituem o caminho de cada rota;
2. **Marcar pontos/troços de interesse numa rota:** o utilizador pode marcar pontos/troços relevantes para uma determinada rota, atribuindo também o respectivo comentário;
3. **Classificação de rotas:** classificação das rotas quanto à sua dificuldade, esta pode ser atribuída, quer pelo utilizador ou pela própria aplicação;
4. **Sugestão de rotas alternativas:** quer o sistema quer os utilizadores poderão ser capazes de sugerir rotas alternativas aos utilizadores;
5. **Mapa com todas as rotas** mapa que contém todas as rotas partilhadas pelos utilizadores;

6. **Gráfico de perfil do percurso:** gráfico que relaciona a elevação com a distância percorrida;
7. **Estatísticas simples:** a aplicação permite visualizar estatísticas como a distância percorrida, velocidade média e inclinação do percurso;
8. **Interacção entre utilizadores:** a aplicação permite e promove a troca de informações entre utilizadores;
9. **Usabilidade:** Classificação qualitativa entre 1 e 5, onde 1 representa muito fraca e 5 muito boa.

Tabela 3.1: Tabela de comparação de aplicações.

Funcionalidade	1	2	3	4	5	6	7	8	9
bikely.com	X						X	X*	4
cycle-route.com	X	X	X				X	X**	3
nycbikemaps.com			X		X		X		2
labikepaths.com								X*	2
sharemyroutes.com	X	X	X			X	X	X*	3
everytrail.com	X		X	X***	X		X	X****	3
mapmyride.com	X		X	X***	X	X	X	X	5

* - Apenas fazer comentários a cada rota;

** - Fazer comentários a cada rota e fórum de discussão;

*** - Sugestões de rotas nas proximidades;

**** - Estilo Twitter;

Como se pode verificar pela Tabela 3.1 comparativa das aplicações analisadas, a aplicação que reúne maior número de funcionalidades aliada a uma usabilidade muito boa é a mapmyride.com. Exceptuando a marcação de troços/pontos de interesse cobre todas as funcionalidades apresentadas. A marcação de troços/pontos de interesse representa uma funcionalidade bastante importante na aplicação FacebikeDC pois, com base nesta marcação será possível representar quais as áreas críticas à prática de ciclismo numa determinada região. No estudo desta e de outras aplicações não foi possível auferir se estas se destinam à agregação de dados das movimentações dos ciclistas. Um outro aspecto a ter em conta, apesar de existirem já algumas ligações deste conjunto de *websites* ao Facebook, é que são aplicações completamente autónomas, fazendo com que a angariação de novos utilizadores seja sempre um desafio. Estes representam um papel importante em matéria de angariação de informação das movimentações. Sabendo do número colossal de utilizadores que o Facebook possui, considera-se esta uma vantagem da aplicação FacebikeDC em relação às demais.

3.2 Aplicações para visualização de dados e extracção de conhecimento

3.2.1 *Safe Road Maps*

O *Safe Road Maps* é uma aplicação *online* que permite visualizar informações relacionadas com a segurança rodoviária em ambiente urbano ou rural. O principal objectivo deste sistema e dos seus serviços analíticos é a sensibilização dos condutores para a problemática dos factores de risco que levam aos acidentes rodoviários, bem como das suas repercussões [22].

Este sistema permite visualizar os acidentes ocorridos nas estradas dos Estados Unidos da América através de *heat maps* interactivos (Figura 3.1). Neste, pode ser escolhida uma determinada área ou percurso e visualizar a informação relativa aos acidentes abrangidos pela selecção feita (Figura 3.2). A informação presente em cada ponto marcado inclui por exemplo quais as causas do despiste, a idade e sexo dos intervenientes e uma vista do local do acidente através do serviço *Google Street View*.

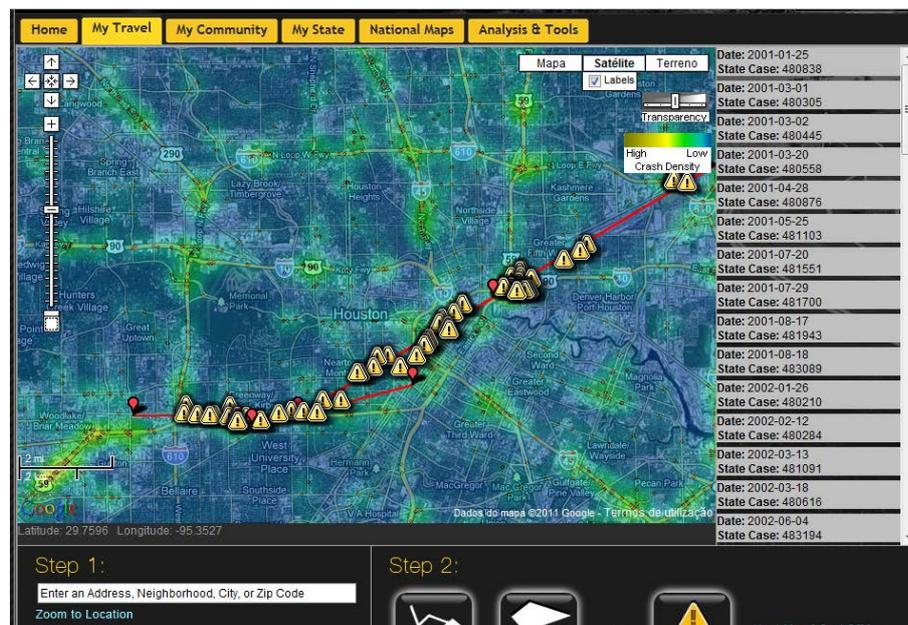


Figura 3.1: Visualização de um *heat map* na aplicação *Safe Road Maps*

3.2.2 hotpads.com

Este *website* tem o objectivo de fornecer informações aos utilizadores sobre o aluguer e venda de imóveis. Estes são apresentados ao utilizador com recurso a um mapa onde são indicadas as suas localizações. É possível também visualizar através de um *heat map* algumas informações relativamente à zona seleccionada no mapa de imóveis (Figura

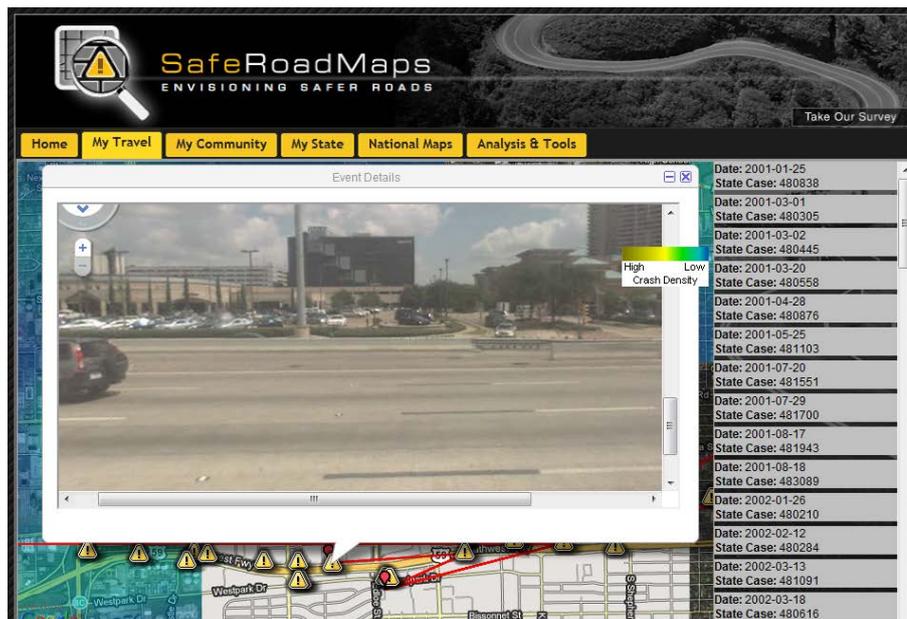


Figura 3.2: Visualização da informação de um acidente na aplicação *Safe Road Maps*

3.3). Os *heat maps* podem ser baseados em várias estatísticas como o rendimento de uma família, a média de idades, a média do valor do aluguer de casas e a densidade de habitantes.

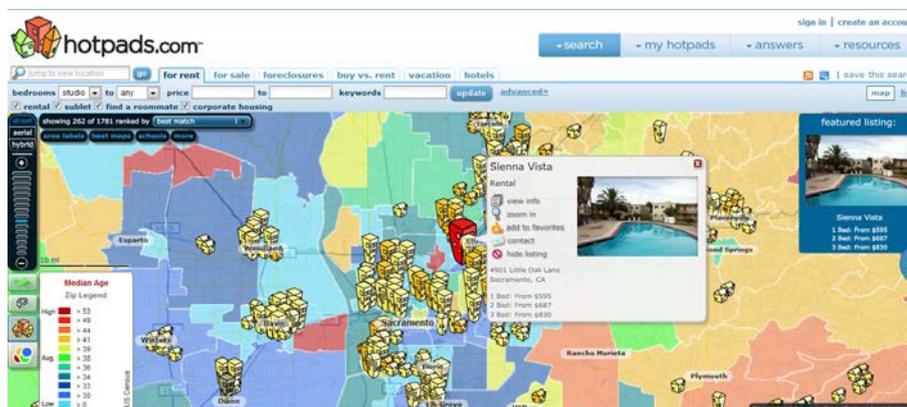


Figura 3.3: Visualização da informação de uma zona de imóveis através da aplicação *UK Heat Map*

3.2.3 *UK Heat Map*

O departamento de energia e alterações climáticas do governo do Reino Unido disponibiliza, no seu *website*, um *heat map* interactivo. Neste mapa é possível ver informações sobre a energia gasta em todo o país (Figura 3.4). Seleccionando uma determinada área é

possível visualizar informações sobre a percentagem de energia que cada tipo de edifícios (doméstica, pública, etc.) gasta. Esta aplicação permite também a selecção de qual o tipo de edifícios a visualizar no *heat map*.

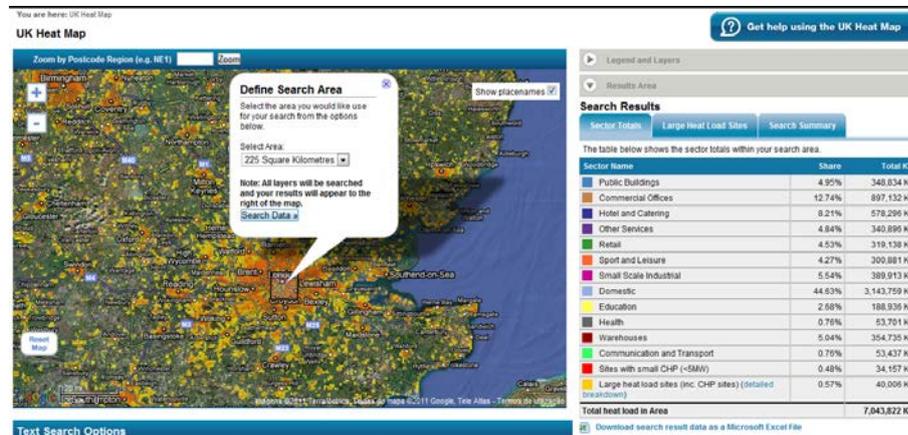


Figura 3.4: Visualização da informação de energia gasta da zona de Londres através da aplicação *UK Heat Map*

3.2.4 Aplicação na visualização e análise de dados extraídos

Embora as aplicações anteriormente apresentadas não estejam directamente ligadas com a prática de ciclismo, estas podem servir de exemplo de como interagir com o utilizador, para que o processo de extracção de conhecimento decorra da melhor maneira possível. A possibilidade que a aplicação *Safe Road Maps* oferece de seleccionar um caminho ou área é bastante útil, permitindo ao utilizador livremente escolher as zonas das quais pretende ver informação. Também nesta aplicação os pontos assinalados onde ocorreram acidentes podem ser vistos através do serviço *Google Street View*. Na aplicação *FacebikeDM*, para casos em que sejam marcados pontos onde este serviço da *Google* esteja disponível, permitir ao utilizador visualizar uma imagem real da zona é bastante útil para saber exactamente qual o interesse daquele ponto. Na aplicação *UK Heat Map* é possível seleccionar qual o tipo de edifício a visualizar. Cada um destes está associado a uma camada no *heat map*, permitindo saber que tipos de edifícios gastam o quê. Transportada a ideia para a aplicação análise de dados, seria bastante útil para o analista escolher, por exemplo, que perfil de ciclista ver num *heat map* de velocidades.

Trabalho relacionado

Capítulo 4

Descrição do sistema

Este sistema, como já referido anteriormente, tem duas funções principais: recolha de dados e sua posterior análise. Sendo assim, este será constituído por duas aplicações *web*. A primeira será a aplicação integrada no Facebook que se destina à recolha dos dados das movimentações dos ciclistas e a segunda será uma aplicação centrada na análise de toda a informação recolhida.

4.1 Funcionalidades propostas para o sistema

As funcionalidades que o sistema deverá albergar, para que os objectivos propostos sejam alcançados, estão divididas em quatro grupos: Sociais, Estatísticas, Centrais e de Gestão. Estas serão descritas nas respectivas secções.

4.1.1 Aplicação de recolha de dados (FacebikeDC)

Nesta aplicação existiram dois tipos de utilizadores: Ciclistas e Administradores. Os utilizadores do tipo Ciclista são os utilizadores comuns do Facebook que utilizam a aplicação integrada nesta rede social. Os administradores deverão também ser utilizadores do Facebook. No entanto, para além das funções dos utilizadores comuns da aplicação, estes terão ainda outras de administração. A Tabela 4.1 apresenta um resumo das funcionalidades e respectivos grupos descritas nas secções que se seguem.

4.1.1.1 Funcionalidades Centrais

Estas funcionalidades são críticas para a aplicação na medida em que é através destas que são recolhidos os dados em bruto, para posterior análise. Sem algumas destas funcionalidades o sistema não poderá funcionar.

- **Autenticar-se na aplicação:** a autenticação da aplicação é feita através da aceitação do utilizador do Facebook para que a aplicação possa aceder aos seus dados pessoais;
- **Criar as próprias rotas através da interface:** através de uma interface *Google Maps* o utilizador poderá marcar os seus percursos directamente sobre o mapa e gerar uma nova rota;
- **Criar as próprias rotas através de logs GPS:** depois de gerado o ficheiro do tipo *GPX* através da marcação da rota efectuada por um dispositivo GPS, o utilizador poderá carregá-lo e a aplicação irá traçar a rota num mapa;
- **Visualizar próprias rotas:** o utilizador deverá poder ver as suas rotas assinaladas num mapa do tipo *Google Maps*;
- **Criar/Editar/Remover pontos de interesse nas rotas:** os utilizadores devem poder criar, editar ou eliminar na própria rota pontos de interesse, associando a estes uma breve descrição. A posição bem como a descrição deverão poder ser editadas;
- **Criar/Editar/Remover troços de interesse nas rotas:** numa determinada rota deverá ser possível ao utilizador assinalar partes desta, adicionando também uma breve descrição.

4.1.1.2 Funcionalidades Estatísticas

Através destas funcionalidades o sistema deverá, com base em estatísticas, mostrar informação relevante para constituir conhecimento para o utilizador.

A aplicação deverá:

- **Sugerir aos utilizadores percursos alternativos:** baseando-se na análise de outras rotas quanto ao tempo, velocidade, declive e dificuldade, o sistema deverá fornecer aos utilizadores percursos alternativos para que estes possam otimizar, de acordo com as suas preferências, as suas rotas;
- **Criar mapa completo com rotas, pontos e dificuldade assinalados:** a aplicação deverá criar um mapa completo contendo as rotas que um determinado ciclista efectuou, assinalando nestas os pontos de interesse bem como o grau de dificuldade da rota;
- **Gerar rankings de ciclistas:** com recurso a várias métricas como distância percorrida, velocidade média e declives, a aplicação deverá criar *rankings* que mostram quais os ciclistas que obtiveram melhor desempenho;

- **Mostrar estatísticas sobre cada rota:** o utilizador deve poder consultar parâmetros como a velocidade, distância, altitude, tempo sobre uma determinada rota;
- **Gerar gráficos do perfil de percurso:** a aplicação deverá gerar um gráfico que relaciona a altitude com a distância de uma determinada rota;
- **Gerar um mapa da rota com indicação do declive:** na aplicação, o utilizador poderá visualizar uma determinada rota em que, com uma determinada escala de cores, são indicadas as diferentes percentagens de inclinação de uma determinada rota. Desta forma será possível ao utilizador ter uma melhor percepção das dificuldades em termos de inclinação que a rota poderá ter.

4.1.1.3 Funcionalidades Sociais

As funcionalidades Sociais têm como principal objectivo promover a interacção entre os ciclistas.

Os Ciclistas deverão poder:

- **Editar o perfil de ciclista:** o utilizador poderá, no seu perfil, indicar quais as suas rotas favoritas, tipos de percursos favoritos, eventos em que irá participar, grupo de ciclistas a que pertence, rotas pessoais e estatísticas sobre os seus percursos efectuados;
- **Editar/Remover as próprias rotas:** deverá ser permitido aos utilizadores editarem as suas rotas de forma interactiva, através de um mapa;
- **Visualizar rotas de outros ciclistas:** cada ciclista tem a opção de partilhar ou não a sua rota com os restantes ciclistas, permitindo que estes visualizem as suas informações;
- **Adicionar comentários a uma rota:** nas rotas que forem partilhadas, cada utilizador poderá adicionar comentários;
- **Sugerir percursos alternativos aos seus amigos:** será possível a um utilizador sugerir percursos alternativos a uma determinada rota, assinalando na rota original as alternativas propostas;
- **Adicionar utilizadores a uma rota:** a cada rota será possível associar utilizadores que tenham participado nela;
- **Criar eventos:** os utilizadores poderão criar, através da aplicação, eventos para os ciclistas, como corridas, encontros, rotas turísticas, etc. Estes eventos serão provenientes do Facebook e integrados na aplicação;

- **Criar grupos de ciclistas:** à semelhança dos eventos anteriormente referidos, os ciclistas poderão criar grupos de ciclistas através da aplicação provenientes do Facebook;
- **Criar grupos de rotas:** será possível aos utilizadores agruparem as rotas em que participam.

A aplicação deverá:

- **Sugerir amigos que tenham características em comum:** o utilizador deverá receber alertas sobre a existência de outros ciclistas que tenham características em comum como semelhança de percursos e *performances*, em termos de velocidade ou distância percorrida.

4.1.1.4 Funcionalidades de Gestão

Estas funcionalidades têm o propósito de que o sistema seja monitorizado, para protecção dos próprios utilizadores.

Através de um módulo de administração, os utilizadores do tipo Administrador deverão poder:

- **Criar/Editar/Remover rotas;**
- **Criar/Editar/Remover utilizadores.**

4.1.2 Aplicação de visualização e análise de dados (FacebikeDM)

A visualização de dados através desta aplicação terá como objectivo principal a extracção de conhecimento sobre o comportamento da comunidade de ciclistas numa determinada área geográfica. A aplicação de visualização dos dados recolhidos através da rede social Facebook deverá gerar mapas de dispersão que serão construídos com base em várias métricas, como a frequência de passagem de ciclistas, dificuldade do percurso, velocidade, declive e horas de trânsito de velocípedes. Idealmente, o utilizador deverá ser capaz de, através da observação dos mapas, responder a questões do tipo "em que zona da cidade será necessário construir ciclo-vias?" ou "em que zona da cidade se deverão construir parques para bicicletas?".

Os mapas gerados terão como base a interface *Google Maps*, na qual serão assinaladas, recorrendo a um código de cores, as zonas que apresentam actividade por parte dos ciclistas. As cores representam uma escala que irá da zona de maior densidade para a de menor. Depois do utilizador escolher uma determinada zona com a ajuda do *Google Maps* integrado num *website*, a aplicação irá subdividir esta em zonas de menor dimensão. O número de subdivisões poderá ser definido pelo utilizador, de acordo com o grau

Tabela 4.1: Quadro resumo das funcionalidades do sistema Facebike

Grupo	Funcionalidade
<i>Centrais</i>	Autenticar-se na aplicação Criar as próprias rotas através da interface Criar as próprias rotas através de <i>logs</i> GPS Visualizar próprias rotas Criar/Editar/Remover pontos de interesse nas rotas Criar/Editar/Remover troços de interesse nas rotas
<i>Estatísticas</i>	Sugerir aos utilizadores percursos alternativos Criar mapa completo com rotas e pontos e dificuldade assinalados Gerar <i>rankings</i> de ciclistas Mostrar estatísticas sobre cada rota Gerar gráficos do perfil de percurso Gerar um mapa da rota com indicação do declive
<i>Sociais</i>	Editar o perfil de ciclista Editar/Remover as próprias rotas Visualizar rotas de outros ciclistas Adicionar comentários a uma rota Sugerir percursos alternativos aos seus amigos Adicionar utilizadores a uma rota Criar eventos Criar grupos de ciclistas Criar grupos de rotas
<i>Gestão</i>	Criar/Editar/Remover rotas Criar/Editar/Remover utilizadores

de detalhe que este deseje. Este deverá poder escolher, também, qual o comprimento da escala a utilizar.

4.2 Arquitectura do sistema

Tendo em conta as características do sistema, foi desenvolvido um possível esquema da arquitectura (Figura 4.1) usada. Podem ser visualizados os diferentes módulos que constituem o sistema, as trocas de informação que são efectuadas entre as diferentes componentes, bem como as tecnologias usadas no decurso da implementação. A arquitectura deste sistema é composta por três camadas principais, nomeadamente camada de visualização, camada lógica e de dados.

4.2.1 Camada de visualização

Através desta camada o utilizador pode interagir com o sistema e visualizar a informação das respectivas aplicações. Cada componente *Front End* utiliza a *API* do *Google Maps* no auxílio à visualização de informação e interacção com mapas geográficos da

Descrição do sistema

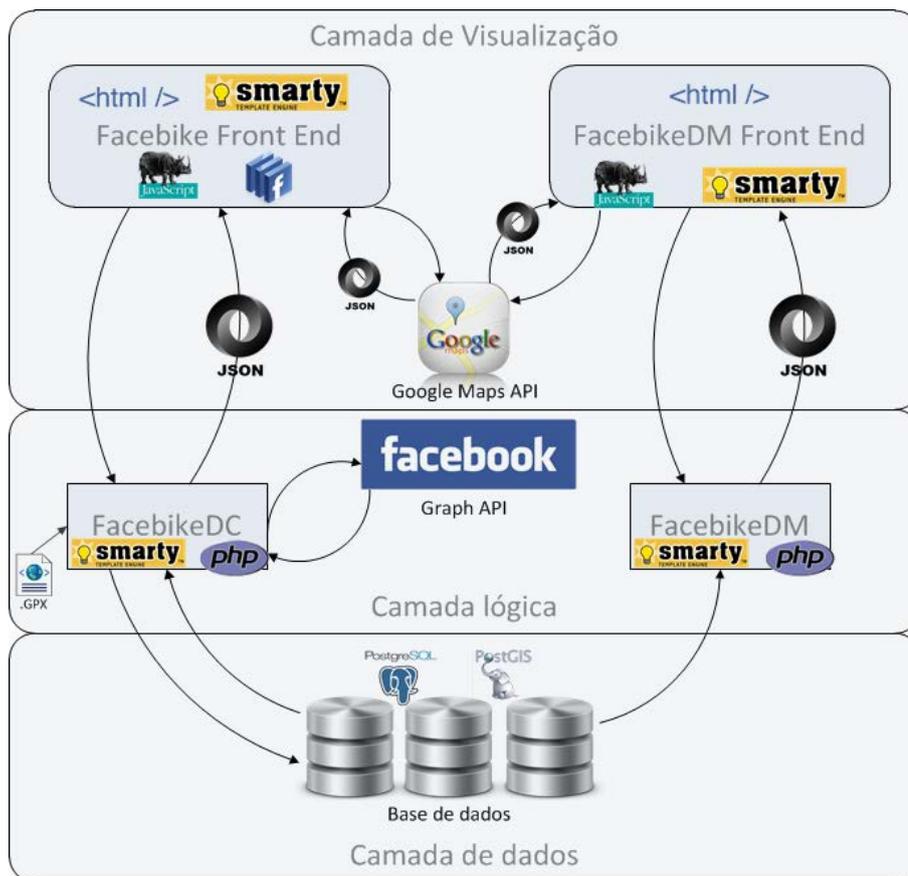


Figura 4.1: Arquitectura do sistema Facebike

terra. Sendo esta camada a que tem contacto directo com os utilizadores, requisitos como a usabilidade, *performance* e fiabilidade têm que ser tidos em consideração para que o utilizador se sinta confortável a interagir com o sistema. No caso da aplicação FacebikeDC, a introdução de dados deve ser o mais aproximada possível à real, produzindo assim resultados mais fiáveis na aplicação FacebikeDM.

4.2.2 Camada lógica

A camada lógica tem a função de estabelecer a ligação entre os dados armazenados, representados na camada de dados, e a camada de visualização, transformando-os para que estes possam ser visíveis de forma gráfica. A aplicação Facebike executa trocas de informação com a *Graph API* do *Facebook*, para aceder às diversas informações dos utilizadores.

4.2.3 Camada de dados

Os dados organizados numa base de dados relacional estão aqui representados. Esta camada executa trocas de informação com a camada lógica através do *Back end* das duas aplicações do sistema. No caso da aplicação FacebikeDM, a informação flui somente num sentido, pois como já referido o propósito desta aplicação é somente a análise de dados já recolhidos.

4.2.4 Tecnologias

No diagrama da arquitectura do sistema da Figura 4.1 é possível observar que tecnologias foram usada em cada uma das componentes do sistema. A linguagem *PHP* é usada somente na camada lógica do sistema pois, graças à biblioteca *Smarty*¹, a lógica de uma aplicação pode estar separada da interface gráfica, dando às aplicações maior escalabilidade e organização. As transacções de informação com a *Graph API* do *Facebook* são feitas através de um *SDK* em *PHP*, desenvolvida pelo *Facebook* para facilitar aos programadores a comunicação com a *Graph API*.

A visualização através da interface *Google Maps* é feita recorrendo à linguagem *Javascript* através da qual foi possível aceder às funcionalidades disponibilizadas pela *API* do *Google Maps* cuja resposta aos pedidos se fez através de objectos *JSON*. Este *standard* foi também usado nas respostas da componente lógica das duas aplicações constituintes deste sistema. Este é um formato mais leve que o próprio *XML* e permite igualmente a transferência de dados com uma determinada organização como por exemplo *arrays*. Os pedidos efectuados desde a camada de visualização para a camada de dados são executados assincronamente, recorrendo a *AJAX*. Desta forma é possível que as trocas de informações decorram sem que sejam necessárias alterações nas páginas que estão a ser visualizadas.

O sistema de gestão de bases de dados utilizado é o *PostgreSQL* que, para além de ser *open source*, permite a utilização de uma extensão que dá suporte a dados geográficos, o *PostGIS*. Este possibilita o armazenamento e processamento deste tipo de dados de forma mais eficaz.

4.3 Modelo de dados

Analisando a problemática da construção da aplicação de recolha de dados, obteve-se o modelo de dados da Figura 4.2. Este pretende representar as relações e os objectos que caracterizam os dados que irão ser processados pelo sistema.

¹Smarty é uma biblioteca de *PHP* para manipulação de *templates* para aplicações *web*

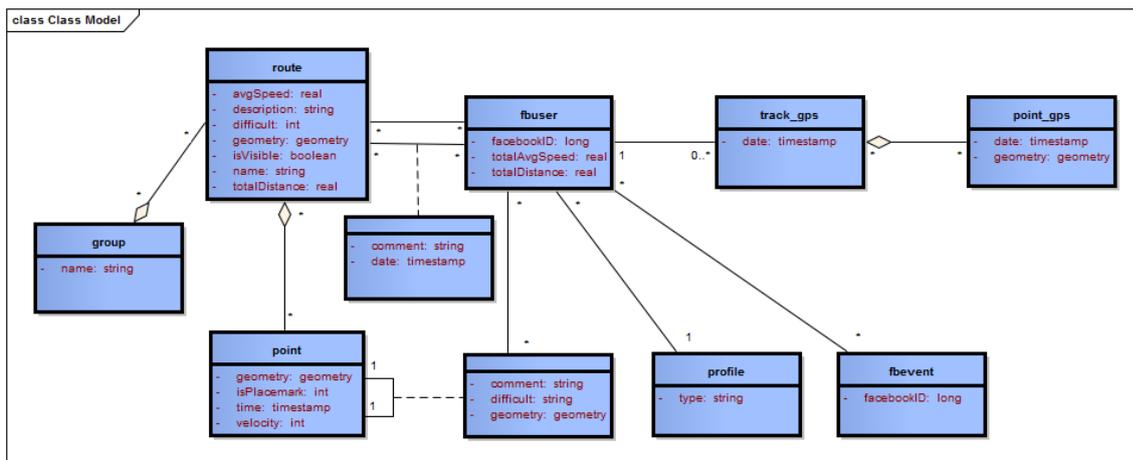


Figura 4.2: Modelo de classes do sistema

4.3.1 Classes

Em seguida é descrito por cada classe qual a sua representação para o sistema.

fbuser: Cada utilizador do Facebook que utiliza a aplicação FacebikeDC.

track_gps: Rotas marcada por um dispositivo GPS ou através da própria interface.

point_gps: Pontos pertencentes às rotas na classe track_gps.

route: Rotas da classe track_gps após o seu de processamento.

point: Pontos pertencentes às rotas em route.

fbgroup: Grupos de rotas.

profile: Tipo de perfil de cada utilizador.

fbevent: Eventos criados pelos ciclistas.

4.3.2 Relações

As relações existentes entre as classes fbuser e route têm o propósito de representar as rotas favoritas de cada utilizador, o criador de uma determinada rota e, finalmente, na classe associação, os comentários que cada utilizador fez às rota. Cada entidade da classe *point* pode-se relacionar com outra entidade do mesmo tipo pois, quando um utilizador assinala numa rota um troço de interesse, este é definido por um ponto inicial e um final. A classe associação que surge desta relação pretende representar os comentários que um determinado utilizador faz ao troço que marcou.

4.3.3 Modelo relacional de dados

O esquema da Figura B.1 representa as relações, atributos e operações existentes entre as várias tabelas da base de dados do sistema implementado. Uma descrição detalhada de cada tabela pode ser encontrada no anexo A.

Descrição do sistema

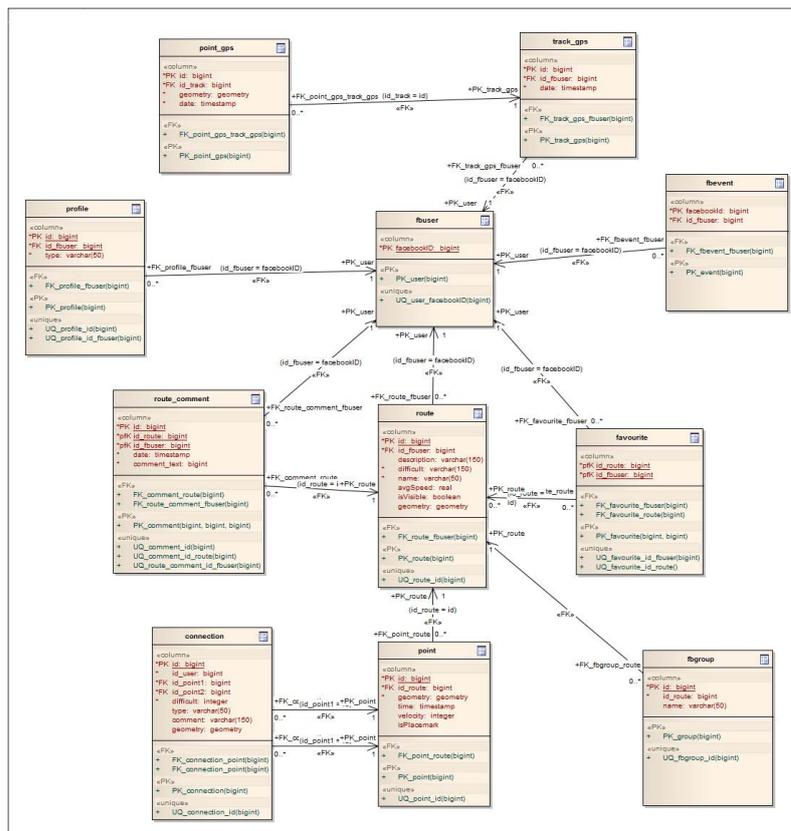


Figura 4.3: Tabelas e relações da base de dados do sistema

4.3.3.1 Geometrias

Para a representação dos dados espaciais sobre as rotas foi usado o tipo de dados espacial *geometry*. Cada *geometry* tem também associado um tipo. No caso do modelo de dados apresentado foram usados os tipos *LINestring* e *POINT*. *POINT* representa um ponto no espaço através das suas coordenadas. *LINestring* é a representação de um conjunto de pontos, formando um caminho. As tabelas *point* e *point_gps* possuem um campo *geometry*, do tipo *POINT*, pois cada uma destas é a representação de todos os pontos constituintes de uma rota. As tabelas *connection*, *track_gps* e *route* têm um campo *geometry* do tipo *LINestring*. Este tipo de geometria pode ser útil, por exemplo, para o cálculo de distâncias entre rotas e intersecções entre estas.

4.3.3.2 Índices

É de prever que as tabelas que contêm os pontos das rotas alberguem um número elevado de registos. Para que o tempo de acesso seja diminuído, os campos *geometry* das tabelas *point* e *point_gps* foram indexados. Os índices criados são do tipo *Generalized*

Search Tree (GIST), que são usados para acelerar buscas em todos os tipos de estruturas de dados irregulares [23].

4.4 Implementação

Em seguida são descritos, por cada funcionalidade, os métodos utilizados na sua implementação e as principais decisões tomadas.

4.4.1 Integração com o Facebook

Para a criação de aplicações integradas no Facebook, está disponível para todos os utilizadores uma plataforma para *developers*². Aqui é possível configurar as ligações para o servidor onde estará alojada a aplicação, inclusão de programadores para a aplicação, entre outras configurações relacionadas com o funcionamento desta. O *Canvas* de uma aplicação pode ser de dois tipos, nomeadamente integrado numa *iframe* ou utilizando a *Facebook Markup Language (FBML)*. A aplicação FacebikeDC é uma aplicação do tipo *iframe* na qual é possível implementar um *website* com resultados aproximados a uma aplicação *web* independente. A escolha recaiu neste tipo de *Canvas* pois a linguagem *FBML* vai, dentro de pouco tempo, deixar de ser suportada nas aplicações do Facebook³. As aplicações integradas no Facebook têm um limite na largura da sua área visualização de 760 *pixels*, o que numa aplicação onde são visualizadas rotas através de um mapa constitui uma desvantagem. No sentido de tentar colmatar este facto, o espaço para a interface *Google Maps* foi maximizado e alguns botões de funcionalidades incluídos na área de desenho do mapa.

4.4.1.1 Autenticação na aplicação

Ao aceder a primeira vez à aplicação Facebike, como em qualquer outra aplicação do Facebook, os utilizadores são questionados a permitir a partilha das suas informações pessoais, bem como executar acções como publicar no próprio mural ou enviar um e-mail elaborado pela aplicação. O processo de autenticação é feito a partir do *SDK* em *PHP*, que faz a comunicação com a *Graph API*. A função *getLoginUrl* gera o *link* que permite ao utilizador aceder à página de aceitação mencionada (Figura 4.4). No caso de o utilizador o permitir, este será redireccionado para a página principal da aplicação e a partir desse momento passa a ser um utilizador FacebikeDC.

²<https://www.facebook.com/developers/>

³Anuncio em: <http://developers.facebook.com/docs/reference/fbml/>

Descrição do sistema

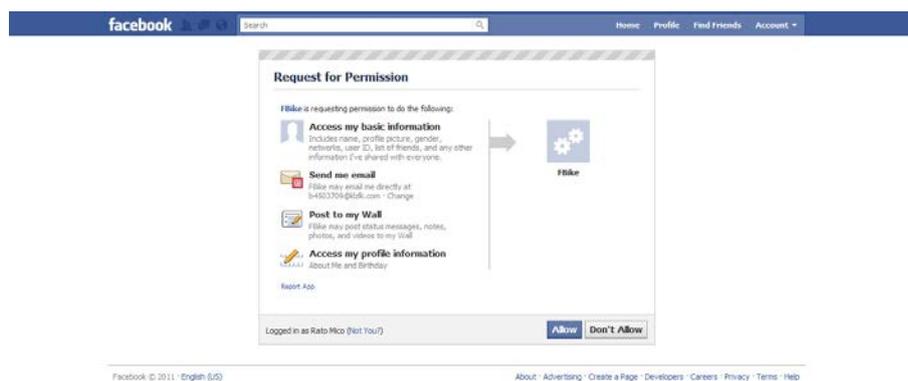


Figura 4.4: Página de autorização de acesso

4.4.2 Criação de rotas

O utilizador poderá criar as suas rotas através do carregamento de ficheiros do tipo *GPX* ou marcando directamente os pontos da interface *Google Maps*. No caso do carregamento de ficheiros, a aplicação analisa-os e acrescenta uma nova rota à tabela *track_gps*, com os campos de velocidade e distância já calculados. Introduce também os novos pontos (*tag <trkpt>*) na tabela *point_gps* com as devidas informações de localização e data/hora caso estejam presentes. Na tabela *point* são também acrescentados os pontos da rota com o aumento de resolução necessário, para os cálculos relacionados com a aplicação *FacebikeDM* (ver Secção 4.4.6). A construção de uma rota através da interface da aplicação é feita recorrendo à *Directions API* do *Google Maps*. Este *Web Service* calcula o caminho mais curto entre um determinado ponto de origem e de destino. O algoritmo utilizado neste serviço retorna também um número mínimo de pontos, por exemplo, se o caminho tiver uma recta esta é definida somente por dois pontos. A inserção das rotas criadas pela interface faz-se de forma análoga a das que são carregadas através de um ficheiro. A criação de rotas através da interface tem uma limitação de oito para o número de pontos de passagem que o utilizador pode assinalar. Esta limitação é imposta pelo serviço *Directions API* ⁴.

4.4.3 Visualização de rotas através do *Google Maps*

Na aplicação *FacebikeDC* os utilizadores podem ver as suas rotas num mapa (Figura 4.5). Estes mapas bem como o desenho da rota, são construídos com recurso à *API* em *JavaScript* disponibilizada pelo *Google Maps*. Através do construtor da classe *Polyline* é possível indicar um conjunto de pontos (no formato de *array*), que constituem um caminho. Os pontos depois são unidos por rectas formando o percurso que foi feito pelo ciclista.

⁴Limites do serviço: <http://code.google.com/intl/pt-PT/apis/maps/documentation/directions/#Limits>

Numa primeira abordagem a este problema foi implementada uma função que construía ficheiros do tipo *KML* com o percurso traçado. Surgiram alguns problemas na visualização destes ficheiros através do *Google Maps*. Por exemplo, quando se alterava o ficheiro a visualizar, a rota que aparecia representada era a anterior. Pelo que foi investigado acerca deste assunto existe um *buffer* que mantém os ficheiros do lado do servidor *Google* para que estes não necessitem de ser lidos várias vezes. Uma questão de melhoria de desempenho, talvez. A solução passa por incluir no ficheiro *KML* uma *tag* que indica qual o tempo de vida deste. No entanto, esta solução não surtiu o efeito desejado. Sendo assim, optou-se pelo método que recorre à *Google Maps Javascript API*.

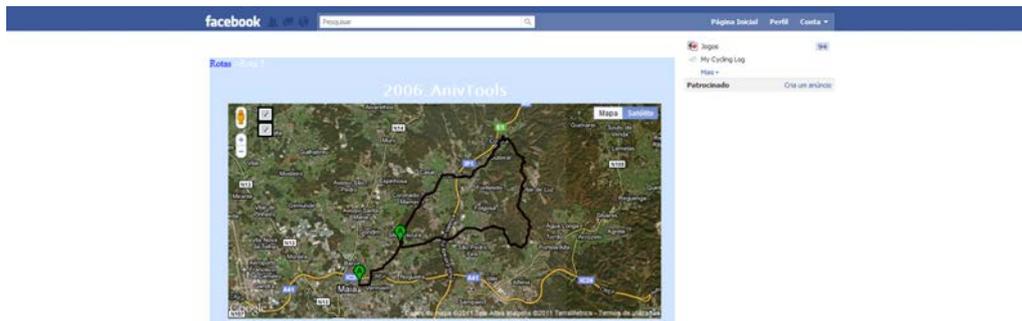


Figura 4.5: Visualização de uma rota através da aplicação FacebikeDC

4.4.4 Cálculo de distâncias e velocidades

Para calcular a distância total que um ciclista percorreu numa determinada rota são somadas todas as distâncias entre cada dois pontos. O cálculo da distância entre dois pontos foi feito recorrendo à fórmula *Haversine*, implementada na linguagem *php*, já descrita anteriormente. Seja P a lista dos pontos de uma rota, ordenados conforme a sua marcação e com a informação sobre a sua latitude e longitude ($P = \{p_1, p_2, \dots, p_n\}$), n o tamanho da lista e DT a distância total, então:

$$DT = \sum_{i=1}^{n-1} Haversine(p_i, p_{i+1}) \quad (4.1)$$

Adicionalmente a extensão *PostGIS* para bases de dados *PostgreSQL*, permite o cálculo destas através da função *ST_Distance* para dados com representação geométrica. Inicialmente, a abordagem para este cálculo passou pelo uso dessa função, mas as distâncias obtidas não correspondiam à distância real. Sendo assim o método de cálculo pela fórmula de *Haversine* foi o utilizado.

Numa fase mais avançada da implementação, ao investigar um pouco sobre este cálculo nas bases de dados geográficas, nomeadamente com a extensão *PostGIS*, foi encontrada uma possível solução para o problemas dos valores errados das distâncias, esta

passaria por, ao invés de utilizar coordenadas geométricas para a representação de pontos, optar pela sua representação de forma geográfica. Uma vez que esta alteração a ser aplicada só poderia trazer vantagens ou não na *performance* da aplicação, optou-se por manter a utilização da fórmula de *Haversine*, para assim poder dar mais atenção a outras funcionalidades também de elevada importância para a aplicação.

No cálculo da velocidade média (*VM*) com que um determinado ciclista efectuou uma rota, são usadas todas as distâncias entre cada sequência de dois pontos bem como o registo sobre a data e hora presente no ficheiro GPX submetido pelo utilizador, sabendo-se assim qual o intervalo de tempo correspondente. Posto isto, basta saber qual o quociente entre a distância e o tempo para calcular a velocidade média da distância correspondente.

Seja P novamente a lista de pontos enunciada em 4.1, T uma lista com a data/hora ($T = \{t_1, t_2, \dots, t_n\}$) da marcação dos pontos da lista P . D representa uma lista de distâncias ($D = \{d_1, d_2, \dots, d_n\}$). Então:

$$d_i = \text{Haversine}(p_i, p_{i+1})$$

$$VM = \left(\sum_{i=1}^{n-1} d_i \right) / \left(\sum_{i=1}^{n-1} t_{i+1} - t_i \right)$$

O método usado para o cálculo das velocidades é um método bastante simplista, tendo em conta que os registos feitos pelos dispositivos de GPS estão sempre sujeitos a erros. Apesar de nos dispositivos GPS mais actuais algumas dessas incorrecções já não se verificarem com tanta acentuação, poderá haver casos em que os *logs* tenham sido feitos por aparelhos mais antigos [24]. Ainda assim, como o objectivo principal desta dissertação não é explorar métodos para reduzir erros de cálculos, mas sim efectuar, através de uma rede social, a recolha de dados e a sua posterior análise, decidiu-se manter o cálculo das velocidades segundo o método já descrito.

4.4.5 Mapa de declives

A implementação deste mapa foi feita com o propósito de gerar ficheiros *KML* que seriam depois visualizados através do *Google Maps*. O mapa de declives que a aplicação deverá gerar, como dito anteriormente, deverá dar a indicação, através de uma escala de cores, da inclinação de um determinado troço. Cada cor corresponde a um intervalo de percentagens de inclinação, em que esta é calculada através do quociente entre a diferença de alturas de dois pontos consecutivos e a distância percorrida entre esses mesmo dois pontos. Seja P a lista dos pontos de uma rota ($P = \{p_1, p_2, \dots, p_n\}$), ordenados conforme a sua marcação e com a informação sobre a sua latitude e longitude, DC a lista de declives ($DC = \{dc_1, dc_2, \dots, dc_n\}$), e A a lista de altitudes ($A = \{a_1, a_2, \dots, a_n\}$) dos pontos em P ,

então:

$$dc_i = ((a_{i+1} - a_i) / \text{Haversine}(p_i, p_{i+1})) * 100$$

Na construção do ficheiro KML são criadas tantas *tags* *<Placemark>* quanto o número de níveis (cores) que a escala tiver. Cada *tag* destas alberga um conjunto de outras *tags* denominadas *<LineString>*. Estas representam um troço que consiste na ligação por rectas dos vários pontos. A cada *Placemark* é atribuída a sua respectiva cor, o que faz com que cada *LineString* seja representada por uma sucessão de rectas com essa mesma cor. Um bom resultado visual deste método, que permita saber com exactidão quais os pontos de maior e menor declive, está bastante dependente da informação disponibilizada pelos aparelhos de GPS quanto à altitude, bem como do intervalo de tempo com que cada ponto é registado no *log* GPS. A Figura 4.6 é um exemplo da aplicação deste método a partir de uma rota de um ficheiro *GPX*.

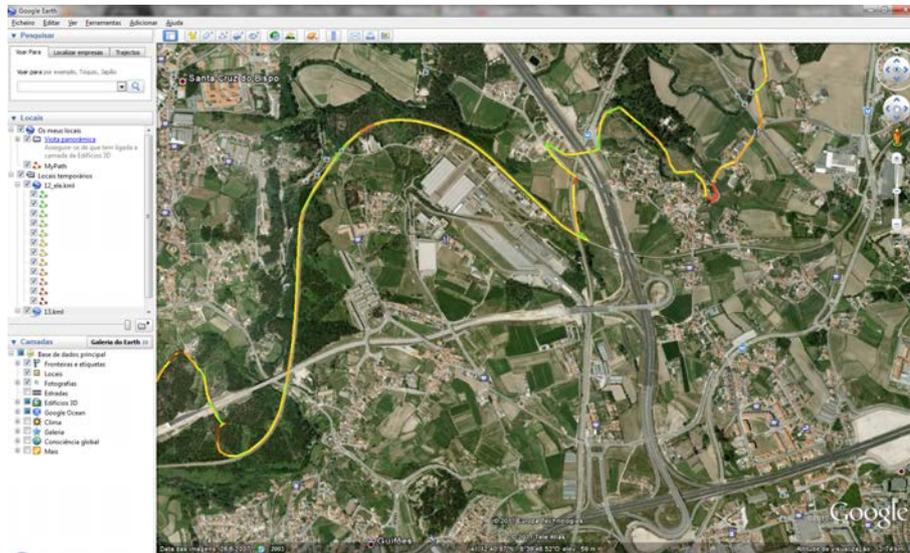


Figura 4.6: Exemplo de um ficheiro *KML* visto a partir do *Google Earth*

Como dito anteriormente existiram algumas dificuldades em visualizar ficheiros do tipo *KML*. No entanto, com as devidas alterações ao código desenvolvido, seria possível gerar as rotas com indicação do declive através da *JavaScript API* do *Google Maps*.

4.4.6 Análise de dados (FacebikeDM)

Das funcionalidades propostas para esta aplicação, a que foi implementada durante esta dissertação foi a visualização de quais as zonas com maior circulação de ciclistas, sendo esta análise baseada unicamente nas rotas marcadas, sem qualquer informação adicional quanto à periodicidade com que estas são percorridas pelos ciclistas. Posto isto,

numa primeira abordagem a este problema, optou-se por efectuar uma contagem do número de pontos que cada rota continha, dentro de cada sub-zona. Rapidamente se chegou à conclusão de que com esta solução os resultados obtidos teriam um grau de fiabilidade muito baixo. Estes estão bastante dependentes do nível de detalhe com que a rota foi traçada. No caso de a rota ser marcada directamente pelo utilizador através da *interface*, o erro de cálculo poderá ser ainda maior, pois como dito anteriormente o número de pontos que representam uma rota é minimizado. A solução pensada foi então o cálculo da distância de todos os troços contidos em cada sub-zona. Para determinar a distância dentro de cada sub-zona, primeiro seriam seleccionados todos os pontos no interior destas. Estando estes já ordenados de acordo com a ordem pelo qual foram percorridos, somente será necessário calcular o somatório das distâncias entre cada conjunto de dois pontos. No entanto, esta abordagem poderia novamente introduzir erros na visualização dos dados, pois poderiam existir zonas que fossem atravessadas por rectas mas que não teriam nenhum ponto marcado nelas, logo a distância desse troço seria nula, o que na realidade não se verifica. Sendo assim, a solução encontrada passou por aumentar a resolução das rotas, dividindo cada segmento em partes iguais, em que a distância máxima entre dois pontos não seria superior a cinco metros. Esta distância foi obtida através da observação de resultados para rotas constituídas por rectas a altitudes baixas (regulada através do *zoom* do *Google Maps*) baixas.

4.4.7 Outras funcionalidades implementadas

Para além das funcionalidades descritas anteriormente, outras também foram implementadas. A aplicação permite aos utilizadores a partilha das suas rotas com os outros ciclistas. Outra funcionalidade social implementada foi a que permite aos utilizadores da aplicação comentarem uma determinada rota.

Através da aplicação é também possível criar, editar e remover pontos/ troços de interesse e seus respectivos comentários. A facilidade com que os utilizadores executam as acções sobre os pontos/troços de interesse é bastante importante para o sistema. É sobre esta informação presente nas rotas que se irá extrair conhecimento relevante para os ciclistas como por exemplo, para gerar rotas alternativas que não passem por um determinado ponto crítico à prática do ciclismo.

A marcação de pontos é feita clicando no percurso da rota para definir a sua localização e adicionando em seguida o respectivo comentário. Para a marcação de troços, o utilizador terá que seleccionar um ponto inicial e um final, adicionando em seguida o respectivo comentário e dificuldade do troço. Todo este processo é feito com o utilizador a interagir com a aplicação unicamente através do *Google Maps* integrado na aplicação, recorrendo à tecnologia *AJAX*.

Descrição do sistema

São gerados também gráficos do perfil do percurso em que é possível saber com exactidão qual o valor de altitude/distância de um determinado ponto de uma rota. Estes gráficos, no entanto, só são gerados para rotas que foram carregadas através de um *log* GPS. Estas contêm no ficheiro *GPX* a informação da altitude, permitindo à aplicação gerar o respectivo gráfico. As rotas geradas através da interface da aplicação não contêm a informação de altitude e o seu gráfico tem sempre o valor 0 para a elevação. O serviço *Elevation API* da *API* do *Google Maps* permite fazer pedidos para o valor da altitude de um determinado ponto. Este serviço tem uma limitação do número de pedidos um pouco baixa para utilizadores do serviço gratuito, que é cerca de 2500 por dia ⁵.

⁵Limites do serviço: <http://code.google.com/intl/pt-PT/apis/maps/documentation/elevation/#Limits>

Capítulo 5

Análise do sistema

Serve o presente capítulo para efectuar a validação do sistema desenvolvido quanto à correcta execução das funcionalidades propostas e implementadas.

5.1 Validação das funcionalidades do sistema

Para validar a correcta implementação do sistema no que toca às funcionalidades implementadas, foram usadas algumas aplicações que apresentam algum grau de fiabilidade no tratamento de dados espaço-temporais.

5.2 Ficheiros de teste

Nos testes efectuados ao sistema foram usados *logs GPS* do tipo *GPX* de rotas marcadas por um grupo de ciclistas. Estes ficheiros correspondem a uma série de rotas percorridas maioritariamente na zona norte de Portugal. Sendo estas representações reais de algumas rotas, constituem um bom caso de teste às funcionalidades implementadas.

5.3 Visualização de uma rota

Para testar a correcta visualização de uma determinada rota através de ficheiros *GPX*, foi usada a aplicação *Google Earth*. Esta permite a visualização deste tipo de ficheiros numa *interface* semelhante à obtida pelo *Google Maps*. Foram usados dois ficheiros de teste: *2007_06_24_RioLeca.gpx* e *2007_07_21_EstVelocipedia40.gpx*. Estes foram carregados na aplicação *FacebikeDC* (Figura 5.2 e Figura 5.4) e no *Google Earth* (Figura 5.1 e Figura 5.3) sendo depois comparada a sua mancha gráfica e localização.

Análise do sistema



Figura 5.1: Vista da rota do ficheiro 2007_06_24_RioLeca.gpx através do *Google Earth*

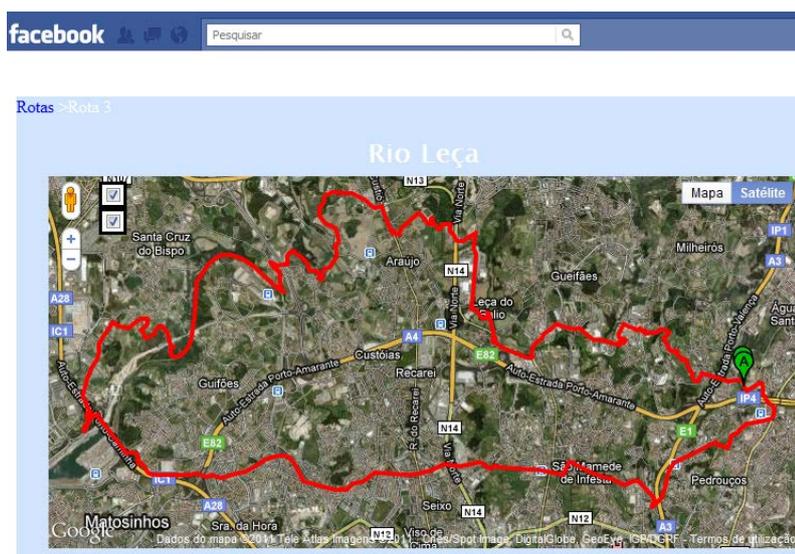


Figura 5.2: Vista da rota do ficheiro 2007_06_24_RioLeca.gpx através da aplicação *FacebikeDC*

5.4 Cálculo de distância e velocidade

O cálculo da distância e velocidade foi verificado recorrendo a três aplicações, nomeadamente o *Google Earth*, a aplicação *web Every Trail* e o *uTrack*. Em cada uma delas foram utilizados os ficheiros 2007_06_24_RioLeca.gpx e 2007_10_14_Valongada-Alvre.gpx, respectivamente 1 e 2 na Tabela 5.1 comparativa de resultados.

Os resultados obtidos (Tabela 5.1) demonstram que o cálculo da distância através da aplicação *FacebikeDC* se aproxima aos valores obtidos através de outras aplicações. A aplicação *Google Maps* é a que se distancia mais dos valores obtidos pelas restantes. No

Análise do sistema

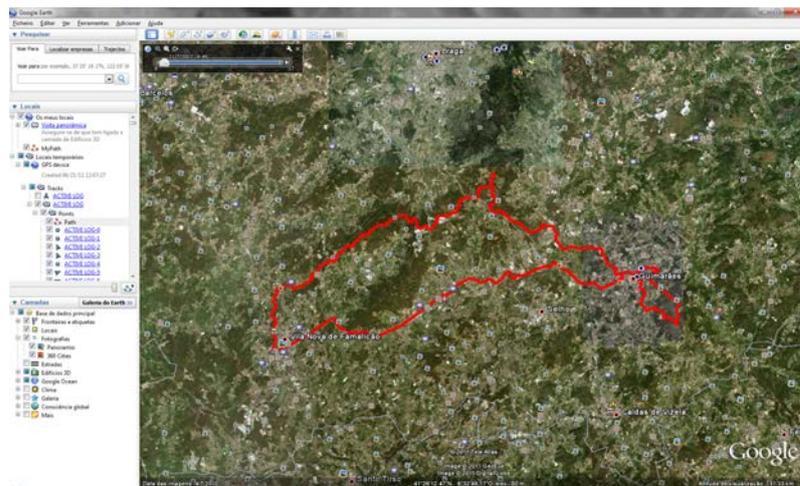


Figura 5.3: Vista da rota do ficheiro 2007_07_21_EstVelocipedia40.gpx através do *Google Earth*

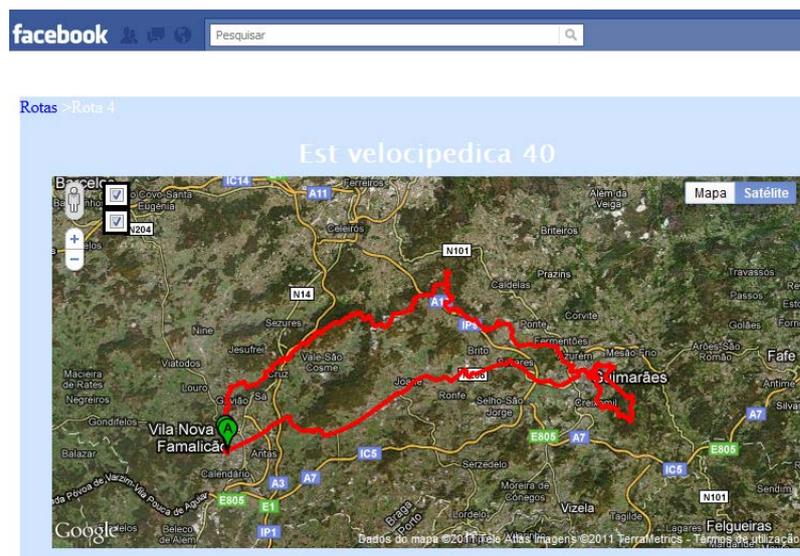


Figura 5.4: Vista da rota do ficheiro 2007_07_21_EstVelocipedia40.gpx através da aplicação *Fa-
cebikeDC*

entanto, como não existe informação de como é efectuado o cálculo da distância nesta aplicação, não se poderá inferir qual a causa para tal diferença. O cálculo da velocidade média apresenta mais disparidades, pois este pode ser efectuado de várias formas. Por exemplo, no *Google Maps* a velocidade é calculada com base no tempo total da rota, esteja o ciclista parado ou em andamento. Para a rota 1, o tempo que dista entre o primeiro ponto marcado e o último é de 3 horas e 29 minutos que, com uma distancia total de 30,6 quilómetros, obtém-se uma velocidade de aproximadamente 8,8 Km/h.

Tabela 5.1: Tabela comparativa de velocidades e distâncias

Aplicação	Rota	Velocidade média (km/h)	Distância total (km)
<i>Google Earth</i>	1	8,8	30,6
	2	8,4	32,2
<i>uTrack</i>	1	13	33,3
	2	12,4	34,3
<i>Every Trail</i>	1	9,33	32,3
	2	8,85	33,8
<i>FacebikeDC</i>	1	9,34	32,6
	2	8,87	34,1

5.5 Gráfico de perfil da rota

O gráfico de perfil de uma rota é um gráfico que estabelece a relação entre distância e altitude. Através deste é possível ter uma ideia de qual o grau de dificuldade do percurso em termos de declive. O gráfico gerado pela aplicação FacebikeDC (Figura 5.5) foi comparado com os gerados por duas aplicações *web*, a *Gps Visualizer* (Figura 5.6) e a *Maplorer* (Figura 5.7).



Figura 5.5: Vista da rota do ficheiro 2007_06_24_RioLeca.gpx através do FacebikeDC

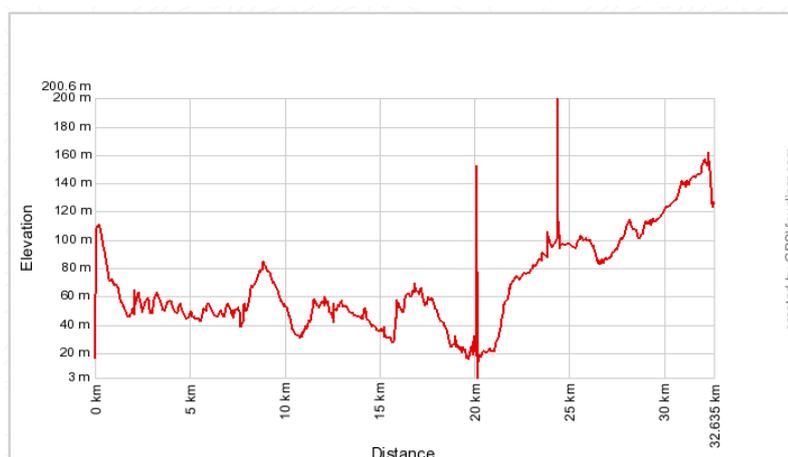


Figura 5.6: Vista da rota do ficheiro 2007_06_24_RioLeca.gpx através da aplicação *Gps Visualizer*

Pela observação dos gráficos gerados, dá para perceber que o da aplicação FacebikeDC se aproxima bastante das das restantes aplicações. Pode-se observar, também, que os

Análise do sistema



Figura 5.7: Vista da rota do ficheiro 2007_07_21_EstVelocipedia40.gpx através do *Maplorer*

gráficos em que o eixo da altitude é mais detalhado, torna-se mais fácil interpretar o grau de dificuldade da rota. Este eixo teve que ser reduzido na aplicação FacebikeDC pela falta de espaço na interface do Facebook.

5.6 Mapa de declives

Na Figura 5.8, no gráfico do *Google Maps* estão os declives reais do trajecto feito pelos ciclistas. A rota representada pelas cores dos seus declives é respeitante ao ficheiro 2007_06_24_RioLeca.gpx.

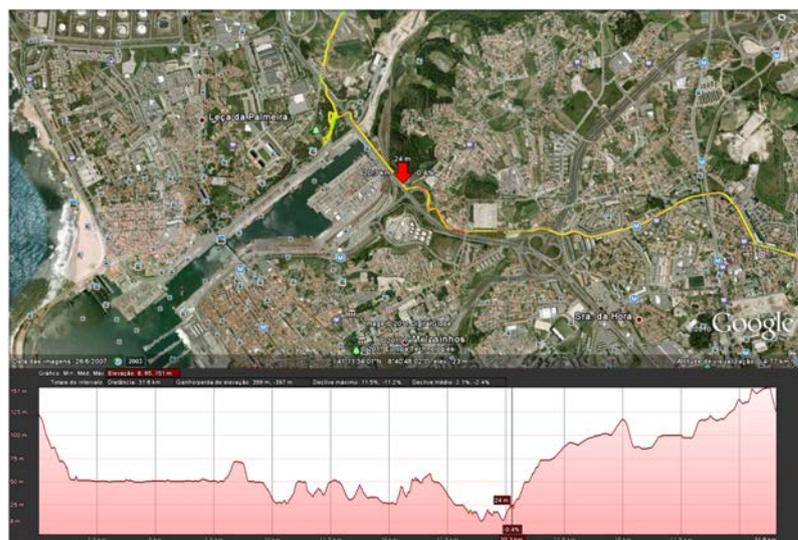


Figura 5.8: Comparação do mapa gerado com indicação através de cores do declive com *Google Maps*

A conformidade destes mapas com os declives reais esta sempre dependente da fiabilidade com que o aparelho GPS faz a marcação de altitudes. Dá para perceber, que no caso desta rota, em grande parte dos casos as cores que apresentam uma tonalidade mais

verde representam descidas. As cores que se aproximam mais dos vermelhos representam subidas. Esta análise é, no entanto, bastante superficial. A Figura 5.9 estabelece a comparação entre os declives reais do terreno e os que o aparelho GPS marcou. Pode-se verificar que existem pequenos erros na marcação da altitude que podem afectar a correcta visualização dos declives.

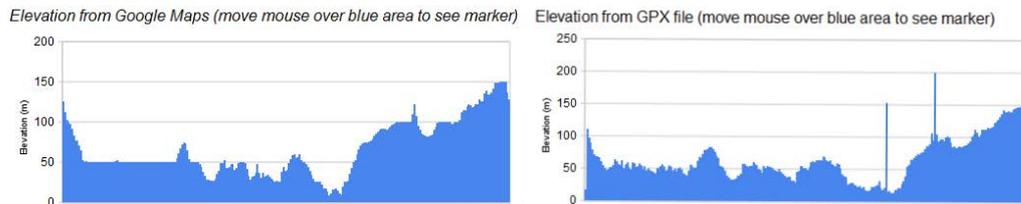


Figura 5.9: Comparação entre o gráfico de perfil gerado através da API do Google Maps e do gerado a partir do ficheiro GPX na aplicação *Maplorer*

5.7 Aplicação de visualização de dados (FacebikeDM)

Para avaliar as funcionalidades implementadas na aplicação FacebikeDM, foi seleccionado um conjunto de rotas de entre os ficheiros de teste. Este conjunto de ficheiros representou um total de 721,2 quilómetros de distância percorrida, num total de 14 ficheiros, calculado através da aplicação FacebikeDC. As rotas marcadas são maioritariamente provenientes da região do grande Porto. A aplicação foi executada com (Figuras 5.10 e 5.11) e sem (Figuras 5.12 e 5.13) a implementação do aumento da resolução das rotas, permitindo assim que sejam visualizadas as diferenças entre cada uma das abordagens ao problema. Todas as rotas foram traçadas no mapa para que assim se possa verificar a correcção das cores de cada sub-zona. A gama de cores utilizada vai desde os tons de verde aos de vermelho, por ordem crescente da percentagem da distância total que essa zona alberga.

Pode-se observar, pelos resultados obtidos nas Figuras 5.10, 5.11, 5.12 e 5.13, que existem mais erros na visualização do mapa de dispersão quando o método de aumento de resolução das rotas não é aplicado. Existem zonas que fazem parte de uma determinada rota que não estão sequer assinaladas. Esta diferença deve-se à inexistência de pontos naquelas zonas em questão quando o método não é aplicado. Pela observação da Figura 5.10 também se pode extrair algum conhecimento. Como, por exemplo, que na zona da Maia existe um grande fluxo de trânsito de velocípedes.

Análise do sistema

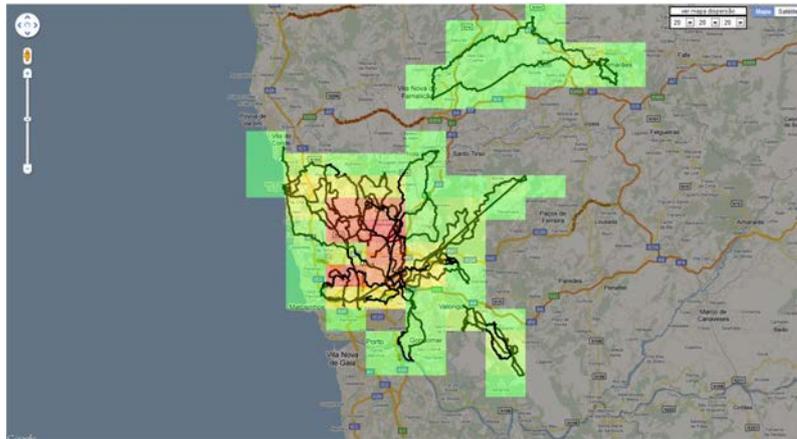


Figura 5.10: Resultado visual da aplicação FacebikeDM com aumento de resolução das rotas

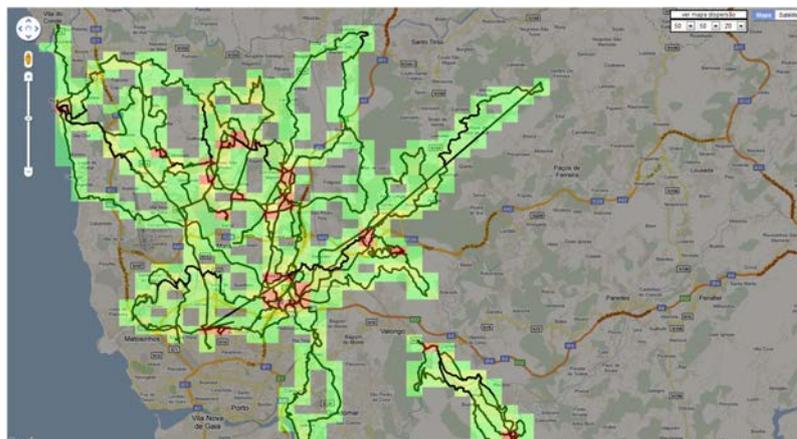


Figura 5.11: Resultado visual da aplicação FacebikeDM com aumento de resolução das rotas

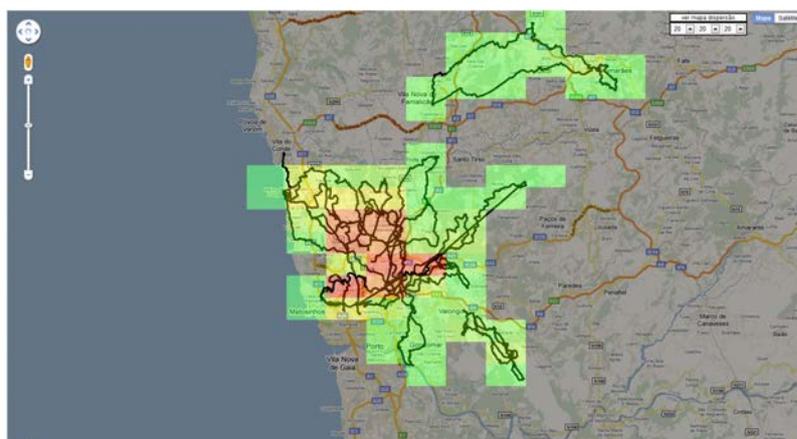


Figura 5.12: Resultado visual da aplicação FacebikeDM sem aumento de resolução das rotas

Análise do sistema

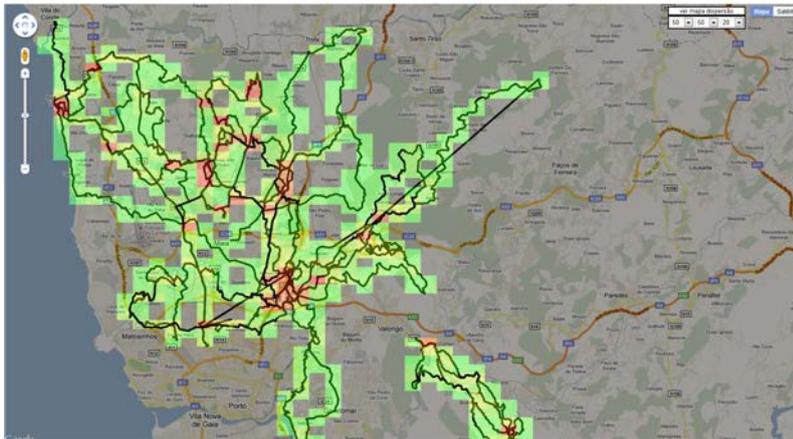


Figura 5.13: Resultado visual da aplicação FacebikeDM sem aumento de resolução das rotas

Capítulo 6

Conclusões e trabalho futuro

O principal objectivo desta dissertação é a agregação e análise de dados das movimentações e comportamentos respeitantes à comunidade de velocípedes. O trabalho realizado teve como principais *outputs* o desenvolvimento de um protótipo funcional, a escrita do presente documento e finalmente a realização de uma prova pública.

Inicialmente foi feita uma recolha de informação relevante para o desenvolvimento do projecto. Foi também analisado algum trabalho relacionado com o intuito de identificar quais as vantagens que o sistema a implementar teria em relação aos demais. Estas tarefas foram documentadas nos Capítulos 2 e 3. A fase seguinte desta dissertação foi a modelação do sistema a implementar. Nesta fase foi projectado o sistema a implementar. Inicialmente foram identificadas as funcionalidades que idealmente o sistema deveria suportar. Posteriormente, foi construído o respectivo modelo de dados e a base dados relacional.

Seguiu-se a fase de desenvolvimento e testes, em que foram implementados os protótipos para as aplicações. À medida que se iam construindo as aplicações foram sendo comparados os resultados obtidos com outras aplicações. Estas duas fases foram documentadas nos Capítulos 4 e 5.

6.1 Satisfação dos objectivos

Embora nem todas as funcionalidades da aplicação tivessem sido implementadas, considero que o objectivo principal da recolha e análise de dados das movimentações geográficas dos ciclistas foi atingido. As funcionalidades implementadas cobrem também as características principais que constituem este sistema, no que toca à interacção social entre utilizadores, recolha e análise dos dados das movimentações geográficas. Considero

também que, através deste sistema, com as devidas alterações enunciadas na Secção 6.3, é possível a recolha e análise de dados junto da comunidade de ciclistas.

6.2 Dificuldades

Ao longo desta dissertação foram encontradas algumas dificuldades que, de alguma forma, influenciaram o resultado final. Inicialmente, a integração com o Facebook teve alguns contratemplos. Apesar do Facebook disponibilizar guiões sobre como efectuar a autenticação dos utilizadores nas aplicações do Facebook, estes estão pouco aprofundados e com escassos exemplos práticos. Pesquisando na *web*, os resultados obtidos eram maioritariamente respeitantes a versões antigas da *Graph API*. Após algumas tentativas falhadas e depois do *SDK* em *php* ter sido actualizado, a autenticação com os utilizadores foi bem sucedida.

O reduzido tamanho disponível em termos de largura das aplicações do Facebook, condiciona a visualização das rotas.

A abordagem inicial para o processamento dos ficheiros envolvidos na aplicação passou pela implementação de um *web service* através das tecnologias *Microsoft .NET*. Foi tomada posteriormente a decisão de que o sistema fosse somente baseado em tecnologias *open source*. Foi necessário que as funções já implementadas fossem modificadas para a linguagem *php*.

Os limites impostos pelos serviços da *Google Maps API* condicionam também a obtenção de melhores resultados do sistema.

Por uma questão de mau planeamento das tarefas a efectuar, o período de modelação foi mais longo do que o que era realmente necessário, o que condicionou o tempo da fase de desenvolvimento.

6.3 Trabalho futuro

Tendo em vista uma primeira *release* deste sistema, seria necessário melhorar e acrescentar algumas funcionalidades, sendo então necessário executar as seguintes tarefas:

- Permitir a edição das rotas, possibilitando aos utilizadores corrigirem os pontos marcados;
- Permitir que a marcação de pontos/troços de interesse suporte múltiplos comentários dos utilizadores;
- Ao carregar uma rota através de um *log* GPS, identificar também os *waypoints*;
- Permitir aos utilizadores adicionar rotas favoritas;

Conclusões e trabalho futuro

- Ao ver a informação sobre cada ponto no gráfico de perfil, mostrar a sua localização no mapa;
- Sistema de *ranking* para a dificuldade de uma rota com base nos votos dos utilizadores;
- *Rankings* de ciclistas;
- Eliminar erros relacionados com os aparelhos GPS;
- Melhorar a visualização das rotas, fazendo *routing* destas ajustando os caminhos traçados aos existentes no mapa;
- Acrescentar outro tipo de métricas (dificuldade, velocidade, etc.) para a análise dos dados recolhidos;
- Efectuar testes de usabilidade;
- Aumentar a segurança da aplicação para utilizadores mal intencionados;
- Quando um determinado utilizador acede pela primeira vez à aplicação Facebike ou cria uma rota nova, publicar no seu mural do Facebook. Desta forma a aplicação poderá ser divulgada aos seus amigos do Facebook.

Numa fase posterior poderia estender-se a aplicação aos dispositivos móveis. Desta forma poderia ser possível, por exemplo, acompanhar um percurso em tempo real através da aplicação.

Conclusões e trabalho futuro

Referências

- [1] Spicycles. Cycling on the rise : Public bicycles and other european experiences, Março 2009.
- [2] Ipsos Reid. City of toronto cycling study : Tracking report (1999 and 2009), Dezembro 2010.
- [3] Portland Bureau of Transportation. Portland bike count report 2010, Novembro 2010. <http://www.portlandonline.com/transportation/index.cfm?c=44671&a=327783>.
- [4] San Francisco Municipal Transportation Agency. 2008 san francisco : State of cycling report, 2008.
- [5] CAN Cycling Advocates Network. Invest in cycling as fuel prices rise, say cyclists, Fevereiro 2011. <http://can.org.nz/media/2011/invest-in-cycling-as-fuel-prices-rise-say-cyclists/>.
- [6] World Health Organization. City survey : Analysis and results, 2011.
- [7] Anthony Mills Gene Bellinger, Durval Castro. Data, information, knowledge, and wisdom.
- [8] Department for Communities e Local Government. *Online Social Networks, Research Report*. Communities and Local Government, 2008.
- [9] Pubudu Kodikara. Info-graphic : The big history of social networking, Janeiro 2011. <http://www.techhamlet.com/2011/01/history-of-social-networking/>.
- [10] Urska Demsar. *Data mining of geospatial data: combining visual and automatic methods*. Tese de doutoramento, Department of Urban Planning and Environment - School of Architecture and the Built Environment Royal Institute of Technology (KTH), Abril 2006.
- [11] M Ankerst. *Visual Data Mining*. Tese de doutoramento, Ludwig-Maximilians-Universitat, Munchen, Germany, 2000.
- [12] Yahaya Bin Abd Rahim. *Spatio-Temporal Data Mining On Moving Objects In DBMS*. Tese de doutoramento, OTB Research Institute for Housing, Urban and Mobility Studies - Delft University of Technology, 2007.

REFERÊNCIAS

- [13] Google. Google maps javascript api v3 reference, 2011. <http://code.google.com/intl/pt-PT/apis/maps/documentation/javascript/reference.html>.
- [14] topografix. Gpx 1.1 schema documentation. <http://www.topografix.com/gpx/1/1/>.
- [15] Communications of the ACM. An interview with edsgar w. dijkstra, Agosto 2010. [entrevista online](#).
- [16] Algotist. Dijkstra's algorithm, 2010. http://www.algotist.com/Dijkstra's_algorithm.
- [17] Ian Wakeman. The a* algorithm, 2011. <http://www.cogs.susx.ac.uk/courses/FP/AstarAlgorithm/AstarAlgorithm.pdf>.
- [18] Nils J. Nilsson. *Artificial Intelligence A New Synthesis*. China Machine Press, 1998.
- [19] Movable Type Scripts. Calculate distance, bearing and more between latitude/longitude points. <http://www.movable-type.co.uk/scripts/latlong.html>.
- [20] Facebook. Facebook developers, Junho 2011. <http://developers.facebook.com/docs/>.
- [21] Google. Google maps javascript api v3 examples. <http://code.google.com/intl/pt-PT/apis/maps/documentation/javascript/examples/index.html>.
- [22] Richard Burkhard Benjamin Schooley Brian N. Hilton, Thomas A. Horan. Saferoadmaps: Communication of location and density of traffic fatalities through spatial visualization and heat map analysis. *SAGE Publications*, 2011.
- [23] Spicycles. Manual postgis : Capitulo 4. usando postgis. http://webgis.com.br/postgis/docs/capitulo4_Usando_PostGIS.htm#id3213068.
- [24] Kowoma. The gps system, Abril 2009. <http://www.kowoma.de/en/gps/errors.htm>.

Anexo A

Modelo de dados

A.1 Tabelas da base de dados

Tabela A.1: Tabela fbuser

Nome	<i>fbuser</i>
Descrição	<i>Nesta tabela está contida a informação relevante sobre cada utilizador da aplicação Facebike.</i>
Atributos	<i>facebookID bigint (Chave Primária): identificador único de cada utilizador do Facebook</i>

Tabela A.2: Tabela track_gps

Nome	<i>track_gps</i>
Descrição	<i>Informação sobre uma rota marcada através de um dispositivo GPS.</i>
Atributos	id bigserial (Chave Primária): identificador único de cada rota name varchar(50): nome atribuído á rota aquando da sua marcação através do GPS date timestamp: data em que a rota foi marcada id_user bigint (Chave Estrangeira): identificador único correspondente ao utilizador geometry geometry: representação geométrica do <i>PostGIS</i> de cada rota
Índices	<i>geometry (tipo: GIST)</i>

Modelo de dados

Tabela A.3: Tabela point_gps

Nome	<i>point_gps</i>
Descrição	<i>Informação sobre cada ponto marcado através de um dispositivo GPS.</i>
Atributos	<p>id bigserial (Chave Primária): identificador único de cada ponto</p> <p>id_track_gps bigserial (Chave Estrangeira): identificador único da rota correspondente ao ponto</p> <p>date timestamp: data em que o ponto foi marcado</p> <p>geometry geometry: representação geométrica do <i>PostGIS</i> de cada ponto</p>
Índices	<i>geometry (tipo: GIST)</i>

Tabela A.4: Tabela route

Nome	<i>route</i>
Descrição	<i>Informação sobre cada rota corrigida pelo algoritmo de routing</i>
Atributos	<p>id bigserial (Chave Primária): identificador único de cada rota</p> <p>id_user bigint (Chave Estrangeira): identificador único da rota correspondente ao ponto</p> <p>date timestamp: data em que a rota foi marcada</p> <p>name varchar(50): nome da rota</p> <p>avgSpeed real: velocidade média com que a rota foi percorrida</p> <p>totalDistance real: distancia total da rota</p> <p>difficult integer: dificuldade da rota</p> <p>isVisible boolean: rota visível ou não para outros utilizadores</p> <p>geometry geometry: representação geométrica do <i>PostGIS</i> de cada rota</p>
Índices	<i>geometry (tipo: GIST)</i>

Tabela A.5: Tabela point

Nome	<i>point</i>
Descrição	<i>Informação sobre cada ponto de cada rota presente na tabela route. Os pontos podem ser os gerados pelo algoritmo de routing ou pontos de interesse marcados pelo utilizador</i>
Atributos	<p>id bigserial (Chave Primária): identificador único de cada ponto</p> <p>id_route bigint (Chave Estrangeira): identificador único da rota da tabela route correspondente ao ponto</p> <p>date timestamp: data em que o ponto foi marcado</p> <p>comment varchar(150): comentário curto ao ponto</p> <p>geometry geometry: representação geométrica do <i>PostGIS</i> de cada ponto</p> <p>isPlacemark boolean: indica se o ponto constitui um ponto de interessa marcado pelo utilizador</p>
Índices	<i>geometry (tipo: GIST)</i>

Modelo de dados

Tabela A.6: Tabela connection

Nome	<i>connection</i>
Descrição	<i>Esta tabela contém a informação relativa aos troços que os utilizadores marcaram como sendo de interesse numa determinada rota.</i>
Atributos	<p>id bigserial (Chave Primária): identificador único de cada troço</p> <p>id_route bigint (Chave Estrangeira): identificador único da rota da tabela route correspondente ao troço</p> <p>id_user bigint (Chave Estrangeira): identificador único do utilizador da tabela fbuser que marcou o troço</p> <p>comment varchar(150): comentário curto ao troço</p> <p>difficult varchar(150): dificuldade do troço</p> <p>geometry geometry: representação geométrica do <i>PostGIS</i> de cada ponto</p> <p>isPlacemark boolean: indica se o ponto constitui um ponto de interessa marcado pelo utilizador</p>
Índices	<i>geometry (tipo: GIST)</i>

Tabela A.7: Tabela route_comment

Nome	<i>route_comment</i>
Descrição	<i>Esta tabela contém a informação relativa aos comentários efectuados a cada rota.</i>
Atributos	<p>id bigserial (Chave Primária): identificador único de cada comentário</p> <p>id_route bigint (Chave Estrangeira): identificador único da rota da tabela route</p> <p>id_user bigint (Chave Estrangeira): identificador único do utilizador da tabela fbuser que efectuou o comentário</p> <p>comment_text text: comentário a uma rota</p>

Tabela A.8: Tabela favourite

Nome	<i>favourite</i>
Descrição	<i>Tabela que representa as rotas favoritas de um utilizador.</i>
Atributos	<p>id_route bigint (Chave Estrangeira): identificador único da rota da tabela route</p> <p>id_user bigint (Chave Estrangeira): identificador único do utilizador da tabela fbuser</p>

Tabela A.9: Tabela fbevent

Nome	<i>fbevent</i>
Descrição	<i>Tabela que contém os eventos do Facebook criados pelos ciclistas.</i>
Atributos	<p>facebookId (Chave Primária): identificador único do Facebook para os eventos</p> <p>id_user bigint (Chave Estrangeira): identificador único do utilizador da tabela fbuser</p>

Modelo de dados

Tabela A.10: Tabela profile

Nome	<i>profile</i>
Descrição	<i>Tabela que contém o tipo de perfil de cada utilizador .</i>
Atributos	id (Chave Primária): identificador único do Facebook para os eventos id_user bigint (Chave Estrangeira): identificador único do utilizador da tabela fbuser type varchar(50): tipo de perfil de um utilizador

Tabela A.11: Tabela fbgroup

Nome	<i>fbgroup</i>
Descrição	<i>Tabela que contém os grupos de utilizador do Facebook criados.</i>
Atributos	id (Chave Primária): identificador único do Facebook para os grupos id_user bigint (Chave Estrangeira): identificador único do utilizador pertencente a este grupo

Anexo B

Esquema do modelo relacional de dados

Anexo C

Script para criação da base de dados

```
CREATE TABLE fbuser (  
facebookID bigint PRIMARY KEY  
);
```

```
CREATE TABLE track_gps(  
id bigserial primary key,  
name varchar(50),  
date timestamp not null,  
avgSpeed real,  
avgSpeedMoving real,  
totalDistance real,  
difficult integer,  
id_user bigint references fbuser facebookID),  
isVisible boolean  
);
```

```
SELECT AddGeometryColumn('track_gps', 'geometry', -1, 'LINESTRING', 3);  
CREATE INDEX track_gps_gix ON track_gps USING GIST (geometry);
```

```
CREATE TABLE point_gps(  
id bigserial primary key,  
date timestamp,  
speed decimal(10,2),  
id_track_gps bigserial references track_gps(id) ON DELETE CASCADE,  
difficult integer,  
comment varchar(50),  
isPlacemark boolean  
);
```

```
SELECT AddGeometryColumn('point_gps', 'geometry', -1, 'POINT', 3);  
CREATE INDEX point_gps_gix ON point_gps USING GIST (geometry);
```

```
CREATE TABLE route (  

```

Script para criação da base de dados

```
id SERIAL PRIMARY KEY,
id_user bigint NOT NULL,
description varchar(150),
difficult varchar(150) NOT NULL,
name varchar(50) NOT NULL,
FOREIGN KEY (id_user) REFERENCES fbuser (facebookID)
);

SELECT AddGeometryColumn('route', 'geometry', -1, 'LINESTRING', 3);
CREATE INDEX route_gix ON route USING GIST (geometry);

CREATE TABLE point (
id SERIAL PRIMARY KEY,
id_track_gps bigserial references track_gps(id) ON DELETE CASCADE,
date timestamp,
velocity integer,
difficult integer,
comment varchar(50),
isPlacemark boolean
);

SELECT AddGeometryColumn('point', 'geometry', -1, 'POINT', 3);
CREATE INDEX point_gix ON point USING (geometry);

CREATE TABLE connection (
id SERIAL PRIMARY KEY,
id_user bigint NOT NULL,
id_point1 bigint NOT NULL,
id_point2 bigint NOT NULL,
id_track_gps bigserial references track_gps(id) ON DELETE CASCADE,
difficult integer NOT NULL,
type varchar(50),
comment varchar(150),
FOREIGN KEY (id_user) REFERENCES fbuser (facebookID),

FOREIGN KEY (id_point1) REFERENCES point_gps (id),
FOREIGN KEY (id_point2) REFERENCES point_gps (id)

);

SELECT AddGeometryColumn('connection', 'geometry', -1, 'LINESTRING', 3);
CREATE INDEX connection_gix ON connection USING GIST (geometry);

CREATE TABLE route_comment (
id SERIAL PRIMARY KEY,
id_route bigint NOT NULL,
id_user bigint NOT NULL,
```

Script para criação da base de dados

```
date timestamp NOT NULL,  
comment_text text NOT NULL,  
FOREIGN KEY (id_route) REFERENCES track_gps(id),  
FOREIGN KEY (id_user) REFERENCES fbuser (facebookID)  
);
```

```
CREATE TABLE favourite (  
id_route bigint NOT NULL,  
id_user bigint NOT NULL,  
FOREIGN KEY (id_route) REFERENCES route (id),  
FOREIGN KEY (id_user) REFERENCES fbuser (facebookID)  
)  
;
```

```
CREATE TABLE fbevent (  
bigint PRIMARY KEY,  
id_user bigint NOT NULL,  
FOREIGN KEY (id_user) REFERENCES fbuser (facebookID)  
);
```

```
CREATE TABLE profile (  
id SERIAL PRIMARY KEY,  
id_user bigint NOT NULL,  
type varchar(50) NOT NULL,  
FOREIGN KEY (id_user) REFERENCES fbuser (facebookID)  
);
```

```
CREATE TABLE fbgroup (  
facebookID bigint NOT NULL,  
id_user bigint NOT NULL,  
FOREIGN KEY (id_user) REFERENCES fbuser (facebookID)  
);
```