

Recolha e tratamento de dados para mobilidade em contexto de cidade inteligente



Submetido por:

José Almeida

Orientado por: Dr. Pedro Pimenta, Prof. João Paulo Leal

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

Índice

Lista de Figuras	ii
Introdução	1
1 Estado da Arte	3
1.1 Programa BaZe	3
1.2 Plano de Trabalho	4
1.3 Tecnologias utilizadas	4
2 Descrição do Trabalho Realizado	7
2.1 Recolha de dados	10
2.1.1 <i>WebScrapping</i>	10
2.1.2 <i>OpenStreetMap</i>	10
2.1.2.1 <i>OverpassTurbo</i>	12
2.1.3 Recolha de horários por paragem	13
2.1.4 Tratamentos de dados	14
2.2 Armazenamento de dados	15
2.2.1 Desenho da base de dados	15
2.3 Disponibilização de dados	16
2.3.1 Construção de uma <i>API</i> em <i>NodeJS</i>	16
2.3.2 Especificação da <i>API</i>	16
Conclusão	18
Referências	19

Lista de figuras

1.1	Plano de Trabalho	4
1.2	Exemplo de consulta no <i>OverpassTurbo</i>	5
1.3	Amostra da Tabela Paragens da Base de Dados em <i>SQLite</i>	6
2.1	Temas e subtemas <i>REOT</i>	9
2.2	Atributos <i>OpenStreetMap</i>	12
2.3	<i>Widget SMSBUS - STCP</i> para horários em tempo real	14
2.4	Transformação das fronteiras do concelho da Maia	14
2.5	Desenho da base de dados	15
2.6	Bibliotecas utilizadas	17

Introdução

A consciencialização ambiental é cada vez mais uma parte importante nos nossos comportamentos e costumes; existe uma motivação global para a rápida adoção de práticas amigas do ambiente, tanto a nível pessoal (comportamentos), como a nível institucional (medidas/apoios).

Desta forma, muitas instituições estão a informatizar os seus serviços para poderem monitorizar e perceber que reformas precisam de ser tomadas em cada setor.

O setor dos transportes, onde se foca este projeto, é responsável por 16.2% das emissões mundiais de gases de estufa, dos quais quase 12% são emitidos por transportes rodoviários[2].

Estes dados permitem perceber de que forma a mobilidade afeta a sustentabilidade e descarbonização.

Assim sendo, e tendo em vista a melhoria das condições ambientais, a Câmara Municipal da Maia criou o programa BaZe[17], no qual está inserido este projeto de estágio.

Entidade Acolhedora

O município da Maia pertence ao distrito do Porto, Região Norte e sub-região da área metropolitana do Porto, tendo uma área de 82.99 km² e uma população estimada de 138.349 habitantes (2019)[10]. É um dos concelhos mais industrializados em Portugal, mas pretende, até 2050, tornar a cidade da Maia energeticamente eficiente, e aproximá-la de uma *'zero carbon community'*[14], melhorando assim a mobilidade e qualidade de vida dos seus munícipes.

Projeto e Objetivos

Este projeto centra-se na recolha, tratamento e disponibilização de dados da rede de transportes públicos da Maia para serem visualizados numa plataforma *online* em tempo real. Os dados pretendidos eram dados de georreferenciação das paragens e linhas da rede de transportes e foram obtidos através de pedidos às *API's - Application Programming Interface*. O armazenamento dos dados foi feito inicialmente com ficheiros *CSV - Comma Separated Values* e mais tarde com *SQLite*, separando assim os dados em três grandes tabelas (paragens, linhas e pontos de referência). A disponibilização foi feita através da construção de uma *API* em *NodeJS* onde o servidor respondia assincronamente a consultas feitas pelo cliente. Desta forma, os dados ficam disponíveis sem haver

perdas de desempenho ou competição por recursos.

Inicialmente, foi-nos pedido para percebermos o conceito de *smart city* e *smart mobility*, o que significa a parte inteligente de uma cidade e que métodos são usados para tornar uma cidade *smart*.

Pretende-se também acrescentar à plataforma de visualização, dados referentes às paragens, linhas, rotas e horários dos autocarros da empresa *STCP*[16] no concelho da Maia. Desta forma queremos também melhorar o acesso à informação por parte dos consumidores, disponibilizando rotas e horários da rede intermodal de transportes da Maia (metro e autocarros). Assim, temos como objetivo conseguir desenvolver o sistema de um ponto ao outro, desde a recolha dos dados até à sua disponibilização, fazendo a interligação entre as diferentes tecnologias usadas. Com a interligação feita, o objetivo seguinte seria automatizar todo o sistema para disponibilizar uma forma simples e rápida de atualizar os dados em caso de alterações à rede de transportes.

Capítulo 1

Estado da Arte

Neste ponto é apresentado o programa onde este estágio se encontra inserido, bem como as tecnologias que nele foram usadas e outras que foram consideradas mas não chegaram ao projeto final.

1.1 Programa BaZe

O programa BaZe - **B**alanco **Z**ero de Carbono, é um programa de ações relacionadas com o conceito de *laboratório vivo*[1], permitindo assim a aplicação, demonstração e apreciação de soluções transversais que promovem a descarbonização enquanto fator determinante no alcance da sustentabilidade. Este programa tem, entre outros, os seguintes objetivos[6]:

- Promover um ambiente urbano sustentável e descarbonizado;
- Fomentar a descarbonização da cidade através da implementação de soluções tecnológicas e ações imateriais que aumentem a eficiência e reduzam o consumo de energia, água e outras matérias;
- Fomentar a demonstração e teste de soluções tecnológicas integradas, em contexto real, que tenham potencial comprovado de ser escaladas para a cidade como um todo;
- Aumentar a conectividade ao nível das tecnologias de informação e comunicação recorrendo à Internet das Coisas (IoT) e sistemas de computação na *Cloud*;

Um dos passos necessários para atingir a descarbonização da cidade é diminuir o número de veículos a circular no concelho. O excesso de veículos na estrada não só aumenta a quantidade de poluentes emitidos, piorando a qualidade do ar, como provoca mais acidentes e diminui, em geral, a qualidade de vida dos cidadãos.

Os transportes públicos têm provado ser uma solução eficaz para estes problemas nos

últimos anos. No entanto, a sua adoção fica condicionada, muitas das vezes e principalmente em meios pequenos, pela falta de acessibilidade à informação por parte dos utilizadores.

A recolha e tratamento de dados referentes a transportes públicos permite monitorizar melhor os meios de mobilidade que um município oferece, e também agregar e disponibilizar de uma forma acessível a informação da rede de transportes de uma determinada localidade.

1.2 Plano de Trabalho

No decorrer do projeto também nos foi pedido para apresentarmos o plano de trabalhos sobre a forma de um diagrama de Gantt. Este consistiria na descrição geral da tarefa a realizar, acompanhado da previsão de tempo que iria demorar a realizar.

Como o projeto sofreu algumas alterações ao longo da sua realização, o plano de trabalho partilhado aqui não corresponde completamente ao trabalho feito. O plano de trabalho começava com a recolha de dados, sendo esta recolha muito mais abrangente e complexa do que aquela que depois se veio a fazer. Métricas iniciais não vieram a ser contabilizadas porque a recolha de dados revelou ser mais complicada do que o esperado. Apesar de a formatação dos dados para o modelo *GTFS - General Transit Feed Specification* não ter sido feita na totalidade por falta de dados, os dados recolhidos preenchem grande parte dos requisitos deste modelo. Por último, a intenção seria automatizar o processo de recolha e exportação para a base de dados através do *Apache Kafka* e de todo o software já existente (conectores e sistemas de visualização de dados).

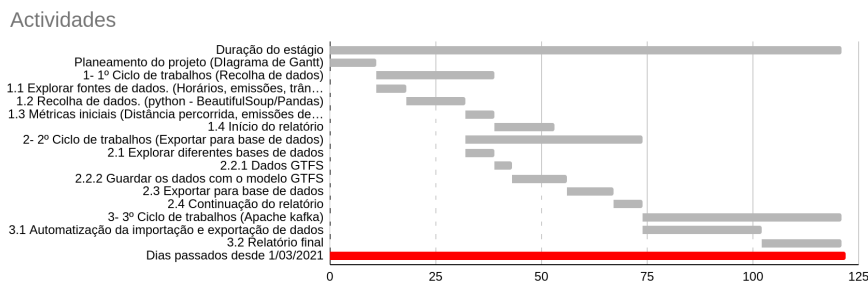


FIG. 1.1: Plano de Trabalho

1.3 Tecnologias utilizadas

O projeto pode ser dividido em três partes e em cada uma delas foram utilizadas tecnologias diferentes.

A primeira parte consistia na georreferenciação das paragens de autocarros e respectivas linhas que estivessem dentro dos limites do concelho da Maia. Para delimitar o concelho da Maia foi utilizado o formato *shapefile* com as delimitações de todos os concelhos de Portugal [12]. Este tipo de ficheiros conseguem descrever no espaço pontos,

linhas e polígonos de forma a representar as estruturas do mundo físico. Estas formas geométricas juntas com atributos associados a cada uma delas formam a representação geográfica dos dados. O formato consiste numa coleção de ficheiros com um prefixo comum armazenados sobre o mesmo diretório, desta coleção há três ficheiros obrigatórios com extensões ".shp", ".shx" e ".dbf", destes ficheiros o mais explorado foi o com a extensão ".shp", este ficheiro guarda os dados geométricos na forma de lista de "records", o ficheiro com extensão ".shx" é um ficheiro posicional, serve para melhorar o acesso ao ficheiro ".shp", o ficheiro ".dbf" é o ficheiro que guarda os atributos associados a cada forma geométrica de modo a complementar a informação geométrica.

Inicialmente foi considerado o uso de *webScrapping* para recolher dados mas depois a falta de estruturação e complexidade da recolha fizeram com que esta técnica tivesse sido abandonada e tivesse optado por sistemas de disponibilização de dados mais fiáveis. Na recolha de dados foram consideradas duas fontes:

Google Developers - Transit[3], a *Google* guarda a informação relativa às paragens e linhas no formato *GTFIS* e disponibiliza, com um custo, uma *API - Application Programming Interface* em *JavaScript* para recolher e visualizar os dados no *Google Maps*.

OpenStreetMap[7], esta plataforma mapeia todo o tipo de locais desde um sinal de trânsito a uma passeadeira. É *OpenSource* e é movida por uma comunidade que contribui para o crescimento e para fiabilidade da informação. A *OpenStreetMap* disponibiliza uma *API* chamada *Overpass API*[8], é uma *API* apenas de leitura que atua como uma base de dados pela internet, o cliente faz uma consulta à *API* e a *API* responde com a informação pedida.

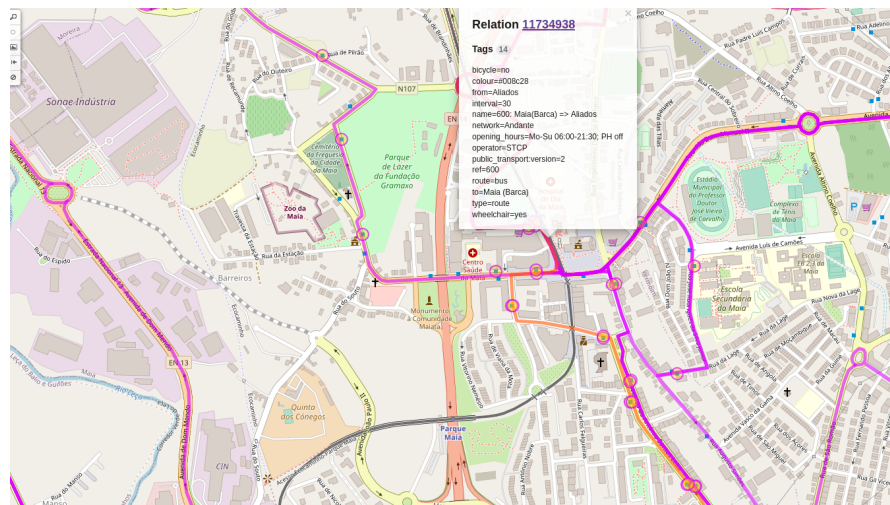


FIG. 1.2: Exemplo de consulta no *OverpassTurbo*

A plataforma escolhida foi a *OpenStreetMap* por ser *OpenSource*, pela fácil acessibilidade aos dados e pela grande comunidade que ajuda na resolução de problemas. Também existe uma aplicação *Web-based - OverpassTurbo* que faz consultas à *Overpass API* e

permite visualizar os dados recolhidos pela consulta, o que facilitou o processo de encontrar a informação desejada[9]. Também foi considerada a plataforma *MoovitApp*[5] mas esta também utiliza a *OpenStreetMap* para recolher dados de georreferenciação. O acesso à *API* foi feito através de *requests Get* escritos em **Python**

A segunda parte consistia no armazenamento da informação. Nesta fase só foi considerado o uso de **SQLite**[15] - esta livreria implementa um pequeno e rápido *SQL database engine* que preenche completamente os requisitos necessários para este projeto. A ter-

	lat	lon	Nome	Referencia
1	41.21652632661527	-8.54150451250296	SONHOS	SONH
2	41.21936228933289	-8.54301056156684	PINHEIRO	PINH2
3	41.21790542268212	-8.54713879886314	5 DE OUTUBRO	SOTR2
4	41.21727714098006	-8.55085747761617	ERMESINDE - B. V.	BVE2
5	41.21877863095163	-8.55283488771321	ERMESINDE (IGREJA)	ERM12
6	41.2165259	-8.5536589	ERMESINDE (ESTAÇÃO)	ERM3
7	41.2132666	-8.5543531	CRUZEIRO	CRZ2
8	41.2110951	-8.5573418	PORTO CARREIRO	PTCR2
9	41.2109869	-8.560882	SAIBREIRAS	SBR2

FIG. 1.3: Amostra da Tabela Paragens da Base de Dados em *SQLite*

ceira e última parte foi a disponibilização da informação presente na base de dados. Para esta tarefa criou-se uma *REST API* em **NodeJS**, com a qual o servidor executa *JavaScript* - a mesma linguagem que o cliente, já existente, executa. A simplicidade da configuração e lançamento do servidor, aliada à performance e à escalabilidade da linguagem, também foram aspectos considerados, assim como a grande demanda de conhecimentos em *NodeJS* por parte das empresas da área.

O *NodeJS* aliava a facilidade de configuração ao desenvolvimento pessoal e por isso foi a escolhida.

Capítulo 2

Descrição do Trabalho Realizado

Este projeto enquadra-se na área de recolha de dados relativos aos transportes públicos que circulam no concelho da Maia. Foram desenvolvidas várias tarefas de preparação antes do começo da parte prática do projeto.

Tal como já foi referido anteriormente, o projeto insere-se no do programa BaZe, Balanço Zero de Carbono, desenvolvido pela entidade acolhedora, a Câmara Municipal da Maia. Este consiste na criação de plataformas de visualização de dados interoperáveis, muitos deles em tempo real, em contexto *smart city*. Foi necessário perceber quais os dados que eram pertinentes neste contexto, e onde se localizavam, para serem posteriormente tratados, disponibilizados e apresentados de forma simples e intuitiva.

Inicialmente, o trabalho realizado foi praticamente todo de pesquisa, utilizando: a norma *ISO37120* [4], o Relatório sobre o Estado de Ordenamento do Território (REOT) [11] e o programa BaZe [17], existente para esclarecer perguntas como:

- O que é uma *smart city*?
- O que é *smart mobility*?
- Que projeto de mobilidade se enquadra melhor nessas definições?

A *ISO37120* é a primeira norma internacional referente a cidades. Esta norma identifica cerca de 104 indicadores de performance (*KPI's* - Key Performance Indicators) distribuídos por 19 áreas de ação diferentes - estes indicadores permitem classificar uma cidade de acordo com a performance dos seus serviços e qualidade de vida dos seus cidadãos. Publicada em 2014 pela *WCCD - World Council City Data*, esta norma já está em prática em mais de 100 cidades espalhadas por 35 países diferentes, atribuindo uma classificação consoante o número de indicadores monitorizados. Apesar de não ser uma avaliação de performance completa, porque não tem em conta o impacto que os dados têm ou não na tomada de decisões por parte das instituições, pode-se dizer que o número de indicadores monitorizados é uma boa métrica de partida para classificar uma cidade como sendo inteligente ou não.

Como já foi referido, esta norma expande-se por 19 áreas distintas, como: economia; educação; ambiente; transporte, planeamento urbano, entre outras. A diversidade de áreas permite a uma cidade controlar muito melhor os seus recursos e serviços, podendo identificar padrões entre serviços independentes que, de outra forma, não se conseguiriam perceber e, assim, melhorar a performance e eficácia na busca pela sustentabilidade. Na área dos transportes, existem quatro indicadores principais:

- Número de quilómetros de transportes públicos de grande capacidade (metro, Avião);
- Número de quilómetros de transportes públicos de baixa capacidade (autocarro, táxi);
- Número anual de viagens em transportes públicos per capita;
- Número de automóveis pessoais per capita.

Estes indicadores permitem avaliar o quanto uma cidade se move de forma inteligente mas não define o conceito de mobilidade inteligente.

”In short, smart mobility is an intelligent transport and mobility network. Smart mobility is the connection of various elements of technology and mobility, a rethinking of the transportation infrastructure used in daily life and business. Not only does this include the use of traditional motor vehicles, electric vehicles, and public transportation systems, but also completely new modes of transportation like on-demand ride sharing services (Uber and Lyft) and carsharing programs” [13]

Uma rede de transportes torna-se inteligente quando associa a tecnologia à mobilidade e, com isso, pode melhorar o serviço prestado ou reinventá-lo por completo. Sistemas de *ridesharing* e *carsharing* permitem reduzir o número de automóveis per capita. Sensores incorporados em veículos podem ser usados para desenvolver sistemas de condução autónoma e assim diminuir o número de acidentes na estrada (indicador secundário *ISO37120*).

Estes sistemas são muito diferentes e respondem a necessidades distintas mas todos eles têm de alguma forma contribuído para a alteração dos indicadores de forma a ir ao encontro de um futuro sustentável.

O *REOT* é um documento bastante diferente que não nos diz o que é uma *smart city* nem o que é *smart mobility*, mas permite-nos ter uma ideia da situação do concelho em todos os domínios que, direta ou indiretamente, influenciam as condições ambientais e de sustentabilidade do território.

Este relatório está dividido em quatro temas principais que se subdividem em subtemas de análise, e que são:

Tema	Subtema
Sócio economia	Dinâmicas Demográficas Estrutura Económica
Mobilidade	Rede de Infraestruturas Movimentos Pendulares Acidentes Rodoviários Transportes Coletivos Modos de Transportes Sustentáveis
Povoamento	Habituação Evolução Urbanística Coesão Urbana Desenho Urbano Equipamentos Públicos
Ambiente	Património Áreas Verdes Arborização em Arruamentos Seccionamento de Ecossistemas Recursos Hídricos Flora Solo Poluição Sonora e Atmosférica Riscos Tecnológicos Infraestruturas

FIG. 2.1: Temas e subtemas *REOT*

Ao longo do relatório conseguimos encontrar vários indicadores que já são utilizados pelo município da Maia para monitorizar os seus serviços. No tema da mobilidade são publicados indicadores como o número de mortes em acidentes rodoviários, o número de metros lineares da rede rodoviária, ou a evolução do número de passageiros por modo de transporte. Quando comparados, percebemos que, de todos os meios de transporte considerados, apenas nos autocarros tem-se verificado um decréscimo constante no número de passageiros.

Sendo o aumento dos preços dos combustíveis uma das principais razões para meios de transporte como o metro terem notado um crescimento de afluência, porque é que o mesmo não se verifica nos autocarros? Algumas razões são apresentadas para justificar a perda de passageiros como:

- A redução de oferta da rede de transportes;
- O aumento das tarifas;
- A situação económica desfavorável na área metropolitana do Porto;
- Registo de fraude a utilização do sistema de transportes;
- Transferência de utilizadores do modo rodoviário para o modo ferroviário.

De forma a combater esta redução de passageiros, contribuir para o desenvolvimento de um indicador primário *ISO37120*, e melhorar a experiência de transporte do consumidor, concluímos que a recolha e disponibilização de dados georreferenciáveis das paragens e linhas seria um projeto pertinente e que poderia enriquecer o programa BaZe no âmbito

da mobilidade inteligente, tendo sempre em vista uma maior sustentabilidade da cidade da Maia. De um ponto de vista pessoal, este projeto foi também vantajoso por ser bastante desafiante e diversificado, com uma utilização de tecnologias atuais.

2.1 Recolha de dados

2.1.1 *WebScrapping*

Como foi dito na introdução, inicialmente, para a recolha de dados, foi considerado fazer *webScrapping* ou seja retirar de uma página *HTML* a informação que era pretendida. Para isso foi utilizado o *Selenium WebDriver* em *Python* para automatizar a navegação no *browser*. Através de pedidos *HTTP* fez-se *webScrapping* ao site da *STCP*[16] e *MoovitApp*[5].

2.1.2 *OpenStreetMap*

O *OpenStreetMap* é um mapa-mundo editável e gratuito criado em 2004 no Reino Unido, motivado pela falta de acesso gratuito e fiável e pelas restrições de utilização dos dados a que estão sujeitos os mapas.

Esta mapa inclui dados relativos a estradas, edifícios, endereços, lojas, pontos de interesse, ferrovias, trânsito, uso do solo, recursos naturais, entre outros - o que lhe dá uma diversidade que torna a plataforma adequada para vários tipos de projetos. O mapa é criado e mantido por quase 5 milhões de utilizadores e mais de 1 milhão de colaboradores de todos os países do mundo, todos utilizando ferramentas e *software opensource* gratuito. Os dados são usados pelas mais variadas pessoas: as comunidades locais, que usam o *OpenStreetMap* para se movimentarem onde vivem; empresas; governos; *developers de software* entre muitos outros.

Este projeto tem uma ligação muito forte à *OpenStreetMap Foundation*, sendo administrada inteiramente por voluntários da Fundação. Existem três componentes básicos do modelo de dados da *OpenStreetMap*, cada um pode ter *tags* associadas, uma *tag* descreve o significado do elemento a que está atribuída contendo toda a informação que não seja a localização. Uma *tag* consiste num par "*key*"="*value*" - a "*key*" é usada para descrever o tópico ou categoria a que pertence o elemento associado e o "*value*" é usado para detalhar o elemento dentro da categoria especificada. O sistema de marcação gratuito do *OpenStreetMap* permite que o mapa inclua um número ilimitado de atributos que descrevem cada característica.

Os três componentes da *OpenStreetMap* são:

- *Nodes* - Um nó representa um ponto específico na superfície terrestre e é definido pela sua latitude e longitude. Cada nó é composto necessariamente por um número identificativo (*id*) e um par de coordenadas. Um nó pode ser usado para identificar

estruturas únicas, como um banco num parque, um sinal de trânsito, uma paragem de autocarro ou um ponto de um caminhom (*way*).

- *Ways* - Um caminho é uma lista ordenada com entre 2 e 2000 elementos que definem uma linha. Os caminhos são usados para representar linhas ou ruas, mas também podem ser usados para representar limites, sendo que seria um caminho fechado em que o primeiro e último nó seriam iguais. Caminhos que necessitem de mais de dois mil nós para serem descritos têm de ser divididos em sub-caminhos sendo preciso criar uma relação que combine os diferentes sub-caminhos.
- *Relations* - Uma relação é uma estrutura de dados multi-propósito que documenta uma relação entre dois ou mais componentes, sejam estes iguais ou diferentes entre si. Estes componentes têm a designação de *relation's members*. A relação é definida pelas suas *tags*, mais particularmente pela *tag: type*. As restantes vão ser adequadas ao tipo da relação.

O esquema de armazenamento para dados relativos a redes de transporte públicos tem vindo a ser alterado com o decorrer dos anos, havendo 4 versões. No entanto, apenas a primeira é mais aceite e comum encontrar na base de dados. Os vários esquemas fazem com que os dados não estejam organizados todos da mesma forma, mas, apesar disso, existem algumas convenções que permitem selecionar exatamente os dados pretendidos. Os serviços de autocarros são mapeados marcando todos os componentes ao longo da rota como membros dessa relação, isto inclui todos os percursos (*ways*) e paragens (*Nodes*). Estas relações podem ser identificadas utilizando a *tag "type" = "route"* para identificar a relação como uma rota, e *"route" = "bus"* para associar a rota a uma rota de autocarros. Para além destes atributos, que são obrigatórios quando se adiciona uma linha, é recomendado ter alguma *tag* que melhor identifica e descreve, a nível local, a linha em questão - para esse efeito é utilizada a *tag "ref" = ""* ou *"name" = ""*.

Existem outros dezassete atributos que podem ser úteis para recolher informação mais detalhada da linha, que são: o operador que está responsável pela linha, o custo, a rede, se está adaptado para cadeiras de rodas, entre outros. Na figura 2.2 (A) podemos ver todas os atributos associadas à linha de autocarros 600 do operador *STCP*. Nesta figura podemos ver que esta linha tem os atributos obrigatórios, *"type" = "route"* e *"route" = "bus"*, e tem também a referência e o nome, que esclarecem melhor de que linha se trata. Existem ainda os atributos adicionais, como o destino, a origem, se tem acesso facilitado a cadeiras de rodas, e também podemos notar que um dos atributos associados à relação é a versão do esquema para transportes públicos, com a *tag "public_transport:version" = "2"*.

Esta relação tem 209 membros, sendo que 45 são paragens (*Nodes*) e o resto sub-caminhos (*Ways*) que, em conjunto, definem o percurso completo da linha. Os *Ways* são estruturas de dados muito simples que guardam um identificador e uma lista de *Nodes* que permitem, através das suas coordenadas, identificar o caminho.

Os *Nodes* são estruturas mais complexas que permitem armazenar mais informação.

FIG. 2.2: Atributos *OpenStreetMap*

Como os autocarros são meios de transporte rodoviários, as paragens são identificadas pelo atributo *"highway" = "bus_stop"* e pelas coordenadas. À semelhança das relações, são usados outros atributos para conseguir associar mais informações à paragem. Na figura 2.2 (B), a paragem "Visconde Barreiros" tem associada a tag *"highway" = "bus_stop"*, a referência com *"ref" = "VBR2"* e o nome com *"name" = "Visconde Barreiros"*. O nome e referência da paragem são depois usados com o *widget SMSBUS* da *STCP* para retornar as próximas passagens da paragem. Para além destes atributos, outros também podem ser considerados úteis para a construção de um projeto de mobilidade, tais como: se existem lugares para sentar nas paragens, caixotes de lixo, se a paragem é abrigada ou não, entre outros.

2.1.2.1 *OverpassTurbo*

O *OverpassTurbo*[9] é uma ferramenta para extrair dados do *OpenStreetMap* baseada na Web, que executa *Overpass API queries* e mostra os resultados num mapa interativo. Esta ferramenta conta com um *"wizard"* que ajuda a construir as consultas com base em palavras-chave.

Esta ferramenta foi muito útil inicialmente para testar *queries* e para perceber quais seriam precisas para recolher os dados que queríamos.

Com os dados das consultas, criámos uma lista de linhas e seus identificadores que depois seriam usados pelo programa *Python* para fazer novas consultas específicas à linha pretendida.

O programa desenvolvido utiliza o interpretador da *API overpass* para selecionar informação da *OpenStreetMap* utilizando os *id's* das linhas, que podem ser vistos no topo da figura (A) em 2.2, e depois explorando os seus membros guardando a informação dos caminhos e nós distinguindo-os através do atributo *"type" = "node"* ou *"type" = "way"*.

Nesta fase o levantamento da informação relativa às paragens foi a que se revelou ser mais trabalhosa, pela falta de consistência na forma que os dados estavam guardados. Depois de os obter, os dados foram guardados em ficheiros *"csv"*.

Foi feito um levantamento das linhas de autocarro que cruzavam o concelho: foram contabilizadas 29 linhas distribuídas por dois operadores, 11 linhas da empresa Maia Transportes e 18 da empresa *STCP*. Destas linhas, foi recolhido o seu identificador, referência, nome e operador responsável.

ID	Ref	Nome	Operador
2826611	10	10: Maia - Praça da República	Maia Transportes
2833825	11	11: Ermesinde - Maia	Maia Transportes
2841575	12	12: Ermesinde (Estação C.P.) - S. P. Avioso	Maia Transportes
...

Dos caminhos, foram recolhidos apenas as coordenadas (lat,lon) dos nós que os constituíam. Das paragens, foi recolhido o identificador, referência, nome, latitude e longitude.

ID	Ref	Nome	lat	lon
2103279279	ESED1	Escola Superior Educação	-8.5977504	41.1808275
2103272930	FEUP1	Faculdade de Engenharia	-8.5985274	41.179192
2169103453	HSJ10	Hospital São João	-8.5997402	41.1830955
...

As relações que descreviam as linhas operadas pelos transportes da Maia não tinham nenhum nó como membro, apenas o caminho que o autocarro faz foi levantado e por isso as linhas deste operador foram descartadas.

2.1.3 Recolha de horários por paragem

Como o *OpenStreetMap* está mais orientado para dados de localização, os dados relativos aos horários das paragens foram conseguidos através do *SMSBUS* disponibilizado pela *STCP*. Esta aplicação recebe o nome e a referência da paragem como argumentos e devolve uma lista de próximas passagens com a linha, direção e tempo até à próxima passagem de um autocarro dessa linha na paragem pedida. A recolha dos horários foi feita no servidor da *API* em *NodeJS* utilizando as bibliotecas *"cheerio"* e *"got"*. A *"cheerio"* é uma biblioteca para *NodeJS* que ajuda a interpretar e analisar páginas *web* usando uma sintaxe parecida à biblioteca *jQuery*.

Começámos por fazer pedidos *HTTP* ao *SMSBUS* com a biblioteca *"got"* - esta ajuda a fazer pedidos *HTTP* e tem dois modos de operação *HTML* e *JSON*. Neste projeto foi utilizado o modo *HTML*, retornando assim uma página *HTML*. Com os selectors do *"cheerio"* extraímos os campos: *"(div[class='floatLeft Linha1'])"* - para devolver a linha, *"(div[class='floatLeft Linha2'])"* - para devolver o destino, *"(div[class='floatLeft*



The screenshot shows a web interface for SMSBUS - STCP. At the top, it says 'HORÁRIOS EM TEMPO REAL'. Below that, it indicates 'Próximas passagens na paragem VISCONDE BARREIROS' and 'Cód. VBR2'. There are buttons for 'Voltar', 'Filtrar', 'Linha', and 'Atualizar'. A table lists the next bus arrivals:

Linha	Destino	Próxima
604	AEROPORTO -	15:03 - 16min
603	MAIA ZOO - M	15:04 - 17min
603	MAIA ZOO - M	15:28 - 41min
604	AEROPORTO -	15:32 - 45min

FIG. 2.3: Widget SMSBUS - STCP para horários em tempo real

Linha2”]]” - para devolver o tempo até à próxima passagem.

2.1.4 Tratamentos de dados

A formatação dos dados ocorreu em dois momentos ao longo do projeto. Primeiramente, os dados foram separados pela sua localização em relação aos limites do concelho. Só os dados referentes a transportes que ocorrem dentro do concelho da Maia precisavam de ser recolhidos, pelo que se recorreu a um *shapefile*[12] disponibilizado por “*dados.gov.pt*” - neste ficheiro os limites do concelho da Maia estão descritos no elemento 213 da lista de concelhos do país. Com a biblioteca *geopandas* lemos os dados do ficheiro conseguindo chegar ao multi-polígono, que descrevia as fronteiras do concelho. Sobre este limite, fizemos uma transformação para expandir a área do concelho de forma a contabilizar as paragens que estivessem perto do concelho da Maia mas não exatamente dentro do concelho. Esta transformação foi feita utilizando o método *buffer* que expande uma série

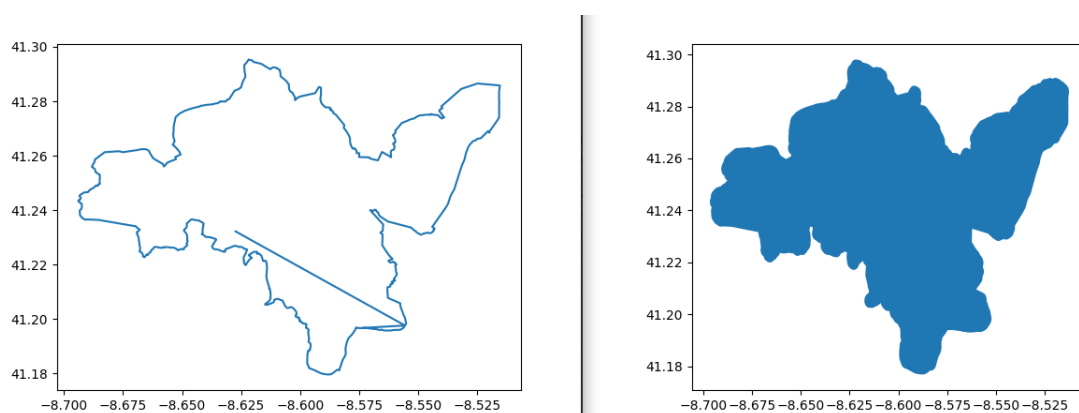


FIG. 2.4: Transformação das fronteiras do concelho da Maia

de pontos para outra que contenha também todos os pontos a pelo menos 0.00269 graus ou 300 metros de distância dos pontos originais.

Numa segunda fase, os dados foram corrigidos à mão, pois o nome e a referência precisavam de estar iguais aos que a *STCP* tinha guardado para podermos usar o *widget* disponibilizado pelos mesmos. Aqui, a solução para o problema não foi a mais indicada mas o produto final ficou como era esperado. Aos dados já recolhidos foi feita a comparação com os dados da *STCP*, fazendo as correções necessárias, e para adicionar novas paragens recorremos ao *google maps* de forma a chegar às coordenadas.

2.2 Armazenamento de dados

2.2.1 Desenho da base de dados

Apesar de o volume de dados não ser muito grande, e tendo em vista a adição de novos dados de outros meios de transporte (aumentando assim o número de acessos), decidimos usar uma base de dados para o armazenamento - ter os dados assim guardados permite que o seu armazenamento seja escalável, mantendo a facilidade e simplicidade na sua alteração e leitura.

Inicialmente, para o desenho da base de dados foram criadas cinco tabelas: uma tabela para as paragens, outra para as linhas, outra para os pontos de referência, e mais duas tabelas que relacionavam as paragens e os pontos de referência às linhas a que pertenciam.

Mais tarde, este desenho foi abandonado, passando para uma versão mais simples em que se eliminou a tabela que relacionava os pontos à linha, adicionando a linha à tabela dos pontos de referência em si. A tabela "Pontos" tem apenas uma chave secundária -

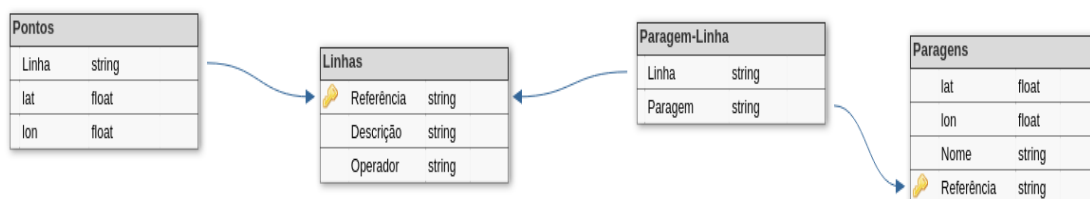


FIG. 2.5: Desenho da base de dados

não tem utilidade distinguir um ponto de outro a não ser que pertençam a linhas diferentes; o atributo "Linhas" é a chave secundária e permite agrupar os pontos por linha. A tabela "Linhas" perdeu o identificador do *OpenStreetMap* pelo que a identificação é feita com a sua referência, mantendo o atributo "operador" e alterando o nome, que agora se chama "descrição".

A tabela "Paragem-Linha" tem duas chaves secundárias de modo a relacionar cada paragem com uma ou mais linhas.

A tabela "Paragens" também perdeu o identificador gerado pelo *OpenStreetMap*, tendo como identificador a sua referência.

TABLE 2.1: Tabela Pontos

Linha	lat	lon
701	41.2456125	-8.5130747
701	41.2456623	-8.5130424
701	41.2456966	-8.5130202
...

TABLE 2.2: Tabela Paragem-Linha

Linha	Paragem
703	SONH
703	PINH2
703	5OTR2
...	...

2.3 Disponibilização de dados

Na disponibilização de dados, um dos objetivos era controlar o acesso aos mesmos e limitar as interações que podiam ser feitas. Para isto, disponibilizar a base de dados não era a opção ideal por não restringir as interações; uma *API*, por sua vez, seria uma melhor opção por só aceitar determinados pedidos. No entanto, ambos têm a possibilidade de controlar acessos. O outro objetivo era oferecer uma forma clara de chegar aos dados, definindo pedidos e respostas para que a transação de mensagens ocorresse de forma estruturada e simplificando o pedido com o trabalho feito no servidor. Desta forma optámos por uma *API* por facilitar o acesso e controlo dos dados.

2.3.1 Construção de uma *API* em *NodeJS*

Para criar o servidor, foi utilizada a ferramenta "*express*", que permite desenhar e construir o *back-end* de aplicações *web* rápida e facilmente. Com a biblioteca *express* podemos criar um objeto "aplicação" que tem já definidos os métodos "*get*", "*post*", "*delete*" e "*set*". Todos os pedidos definidos são pedidos "*get*" e assíncronos, mudando o caminho onde a função é chamada.

Para a conexão com a base de dados em *NodeJS* foi utilizada a biblioteca "*knex*": esta biblioteca inclui "*SQL query builders*" para quase todas as bases de dados baseadas em *SQL*. Com esta biblioteca é mais simples conectar e interagir com a base de dados, sendo preciso apenas usar o nome da tabela e as palavras-chave de *SQL* como métodos ("*select()*", "*where()*").

2.3.2 Especificação da *API*

A informação que queríamos tornar disponível era relativa às linhas, percursos, paragens e horários. Para isso, desenhámos algumas funções que automatizavam as consultas que, à partida, seriam mais frequentes. A *API* ficou com 6 pontos de entrada:

1. lista de linhas guardadas na base de dados;
2. lista de paragens guardadas na base de dados;
3. lista de paragens para uma dada linha;

4. paragem para uma dada referência;
5. lista de pontos de referência para uma dada linha;
6. lista de próximas passagens para uma dada paragem;

A função um devolve um objeto *JSON* com uma lista com todas as linhas que o sistema guarda. Em cada linha existe uma referência, uma descrição e o operador. Esta função seleciona tudo com, ".select("*")", da tabela "Linhas" com, "knex("Linhas")"2.6 (a) e devolve ao método "get" a lista com as entradas da tabela. Este método, por sua vez, responde ao pedido com um objeto *JSON* com a lista das entradas da tabela. A função dois devolve todas as paragens que o sistema guarda.

A função três recebe o identificador da linha e devolve uma lista de referências de paragens que estão associadas a essa linha. Esta função recebe o identificador no caminho que chama o método "get" e seleciona todos as entradas da tabela "Paragem-Linha" cujo atributo "Linha" seja igual ao identificador recebido com, "knex("Paragem-Linha").where("Linha",linha).select("*")"2.6. (b) A função quatro o recebe a referência da paragem e devolve toda a informação relativa à mesma. A quinta função recebe o identificador da linha e devolve uma lista de pontos de referência usados para delinear o trajeto que um autocarro daquela linha percorre.

A sexta função recebe a referência da paragem e devolve uma lista de próximas passagens. Primeiro recolhe da base de dados o nome da paragem a que referência está associada e depois com o nome e a referência usa o *widget STCP* para recolher as próximas passagens. Cada passagem é composta pela referência da linha, o destino e, por fim, o tempo até à próxima passagem.

```
//retorna todas as linhas
app.get("/linhas", async (req, res) => {
  const linhas = await db.getLinhas();
  res.status(200).json({ linhas });
});
//retorna todas as paragens
app.get("/paragens", async (req, res) => {
  const paragens = await db.getAllParagens();
  res.status(200).json({ paragens });
});
//retorna lista de paragens da Linha com ID = id;
app.get("/paragens/:linha", async (req, res) => {
  const paragens = await db.getParagens(req.params.linha);
  res.status(200).json({ paragens });
});
```

(a) Express app

```
function getLinhas() {
  return knex("Linhas").select("*");
};
function getAllParagens(){
  return knex("Paragens").select("*");
};
function getParagens(linha){
  return knex("Paragem-Linha").where("Linha",linha).select("*");
};
function getParagem(ref){
  return knex("Paragens").where("Referencia",ref).select("*");
};
function getPontosRef(linha){
  return knex("Pontos").where("Linha",linha).select("*");
}
```

(b) Knex

FIG. 2.6: Bibliotecas utilizadas

Conclusão

Passando agora para uma reflexão sobre a minha experiência pessoal ao longo deste processo, posso dizer que me foi possível pôr em prática muitos dos conhecimentos que tenho vindo a aprender ao longo do curso, bem como utilizar novas ferramentas com as quais nunca tinha trabalhado antes. Foi bastante proveitoso passar por todas as fases de pesquisa sobre o tema, de recolha de dados e informações, do tratamento e armazenamento dos dados e, depois, da disponibilização dos mesmos. A par disto, tive a oportunidade de experimentar diversas ferramentas, de perceber quais as vantagens de determinadas sobre outras, etc.

Nem todos os objetivos propostos foram atingidos, não existe forma automática e autónoma de replicar o trabalho desenvolvido, apenas os autocarros *STCP* entraram no sistema de dados e não existe uma alternativa local aos dados dos horários das linhas. Apesar destes pontos negativos estou contente com o trabalho realizado face à complexidade do projeto. Creio que o projeto ganha muita força devido àquilo que o move na sua essência. Estando a cidade da Maia localizada num dos dois grandes centros industriais e comerciais do país, o que faz com que tenha um grande número de pessoas que precisam de se deslocar diariamente em transportes públicos, creio que é de uma grande importância as preocupações com a sustentabilidade e com a melhoria da qualidade de vida dos cidadãos na sua rotina quotidiana. Assim, fico bastante contente por saber que o meu trabalho pode contribuir para que uma pequena parte do dia-a-dia de uma cidade, como andar de autocarro por exemplo, se torne mais simples, acessível, e sem obstáculos.

Pensando ainda numa perspetiva de futuro, caso venha a continuar com este projeto, creio que o próximo passo será mudar a fonte de dados para uma mais fidedigna, de modo a não ser preciso corrigir os dados, adicionar outros meios de transporte à base de dados e expandir a *API* para que o cliente existente possa comunicar com os dados apenas por um sítio.

Referências

- [1] Balanço Zero de Carbono. <https://www.cm-maia.pt/pages/1912>, Acedido pela última vez a 29 de Junho, 2021.
- [2] Emissões de gases de efeito de estufa por setor. <https://ourworldindata.org/emissions-by-sector>, Acedido pela última vez a 29 de Junho, 2021.
- [3] *Google Transit API's*. <https://developers.google.com/transit/>, Acedido pela última vez a 29 de Junho, 2021.
- [4] *ISO37120*. https://share.ansi.org/ANSI%20Network%20on%20Smart%20and%20Sustainable%20Cities/ISO%2B37120-2014_preview_final_v2.pdf, Acedido pela última vez a 29 de Junho, 2021.
- [5] *MoovitApp*. https://moovitapp.com/porto-1904/poi/Maia/f/en?fll=41.234613_-8.624074, Acedido pela última vez a 29 de Junho, 2021.
- [6] BaZe - Objetivos. <https://www.cm-maia.pt/baze/objetivos>, Acedido pela última vez a 29 de Junho, 2021.
- [7] OpenStreetMap. <https://www.openstreetmap.org/#map=15/41.2321/-8.6220&layers=T>, Acedido pela última vez a 29 de Junho, 2021.
- [8] *Overpass API*. https://wiki.openstreetmap.org/wiki/Overpass_API, Acedido pela última vez a 29 de Junho, 2021.
- [9] *OverpassTurbo*. <http://overpass-turbo.eu/s/18XQ>, Acedido pela última vez a 29 de Junho, 2021.
- [10] Pordata - Município da Maia. <https://www.pordata.pt/Municipios/Quadro+Resumo/Maia-253316>, Acedido pela última vez a 29 de Junho, 2021.
- [11] *REOT*. https://www.cm-maia.pt/cmmaia/uploads/writer_file/document/2186/REOT_2018_07_13.pdf, Acedido pela última vez a 29 de Junho, 2021.
- [12] *Overpass API*. <https://dados.gov.pt/en/datasets/concelhos-de-portugal/>, Acedido pela última vez a 29 de Junho, 2021.

-
- [13] What is smart mobility and why is it important? <https://www.verizonconnect.com/resources/article/smart-mobility/>, Acedido pela última vez a 29 de Junho, 2021.
 - [14] SPARCS. <https://www.sparcs.info/cities/maia>, Acedido pela última vez a 29 de Junho, 2021.
 - [15] *SQLite*. <https://sqlite.org/index.html>, Acedido pela última vez a 29 de Junho, 2021.
 - [16] STCP. <https://www.stcp.pt/pt/viajar/>, Acedido pela última vez a 29 de Junho, 2021.
 - [17] P. Pimenta. Apresentação do projeto BaZe. https://docs.google.com/presentation/d/1_Ikx0fc9L2KFjMFuVCS019yJhB5bdFInhSMP4J3ji0U/edit?usp=sharing, Acedido pela última vez a 29 de Junho, 2021.