



Análise em tempo real de dados com bases de dados não relacionais

JORGE ABEL DA COSTA CORREIA

novembro de 2016

Análise em tempo real de dados com bases de dados não relacionais

Jorge Abel da Costa Correia

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Isabel de Fátima Silva Azevedo

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, Outubro 2016

Dedicatória

Dedico este projeto aos meus pais que me acompanharam em todo o meu percurso acadêmico e profissional, prestando sempre o seu apoio. Dedico também à minha noiva, cujo apoio e suporte emocional foram fundamentais para a conclusão deste ciclo.

Resumo

Os sistemas de informação são uma peça fundamental no controlo e gestão de vários modelos de negócio. Através deles, torna-se possível a monitorização o controlo e gestão à distância. No caso de empresas de logística, a localização de uma encomenda fica tão fácil quanto o acesso a um dispositivo com acesso à internet. Ainda no contexto de uma empresa de logística, a monitorização em tempo real de motoristas é uma importante ferramenta no processo de tomada de decisão.

A Jumia Services disponibiliza um conjunto de serviços de logística e, a par deste negócio, desenvolve *software* para a gestão dos serviços prestados. Na sua suite de produtos conta já com soluções de *software* que vão desde a gestão de stock em armazém até ao gestão de frotas.

Neste projeto foi realizado um trabalho de investigação com o objetivo de identificar a melhor solução para o armazenamento e acesso a informação, no conjunto de tecnologias utilizadas pela Jumia Services na sua suite de *software*. No seguimento da análise de desempenho, foi desenvolvida uma prova de conceito para a monitorização de motoristas.

O desenvolvimento deste projeto foi um importante contributo para a escolha de tecnologias a adotar no desenvolvimento de soluções de tracking de motoristas e encomendas.

Palavras-chave: Desempenho, Monitorização de motoristas.

Abstract

Information systems are an essential component in the managing and control of a large variety of business models. These systems brings the ability to remotely control and monitor. In the case of logistics companies, get the location of an order is as easy as access a device with internet connection. Also, in the context of a logistics company, real time drivers monitoring might be an essential tool in the decision making process.

Jumia Services provides a range of logistics services. Along with this business is being developing a suite of software products to be used in the services management. Its suite of products already has a wide range of solutions, from the managing of warehouse stock to the fleet management.

The aim of this project is to, considering two different database systems used in the Jumia Services software's, identity the best solution to storage and access information. Following the analysis, it was developed a proof of concept to track drivers.

This project was an important contribution for choosing the technologies to be used in the developing of a track and track software.

Keywords: Performance, Drivers tracking

Agradecimentos

Nesta fase final de mais um ciclo do meu percurso académico, faço questão de agradecer a todas as pessoas que de alguma forma fizeram parte dele.

Agradeço à professora Isabel Azevedo por toda a disponibilidade e compreensão durante todo este percurso. Agradeço também ao supervisor Hugo Conceição, também meu superior hierárquico e amigo, por todo o apoio prestado.

Por fim, gostaria também de deixar uma palavra de agradecimento a todos os meus colegas no Porto Tech Center, que prestaram todo o apoio técnico necessário e me ajudam diariamente a crescer profissionalmente.

A todos, MUITO OBRIGADO!

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Problema	3
1.3	Objetivo e prova de conceito	4
1.3.1	Avaliação de desempenho de bases de dados	4
1.3.2	Desenvolvimento de aplicação protótipo	6
1.4	Abordagem metodológica	7
1.5	Valor de negócio	8
1.5.1	Modelo de negócio de CANVAS	9
1.6	Estrutura da tese	10
2	Revisão bibliográfica	11
2.1	Considerações gerais	11
2.1.1	Bases de dados relacionais	11
2.1.2	Bases de dados não relacionais	13
2.2	PostgreSQL e PostGIS	15
2.2.1	Representação de dados	15
2.2.2	Tipos de dados	19
2.2.3	Funções de acesso aos dados	20
2.2.4	Operadores	20
2.3	ElasticSearch	21
2.3.1	Tipos de dados	21
2.3.2	Funções	22
3	Desempenho	23
3.1	Ambiente	23
3.2	Modelo de dados	23
3.3	Universo de dados	27
3.3.1	Identificação das cidades	27
3.3.2	Desenho de rotas	28
3.3.3	Simulação das viagens	29
3.4	Análise de desempenho	30
3.4.1	Escrita	30
3.4.2	Leitura	33
3.5	Análise do espaço utilizado pelas bases de dados	38
3.6	Conclusões	39
4	Rastreo de motoristas	41
4.1	Arquitetura	41

4.1.1	Controlo de acesso	44
4.2	Requisitos	45
4.2.1	Localização de motoristas	46
4.2.2	Identificação de parceiros mais próximos	46
4.2.3	Encontrar motorista.....	47
4.3	Design.....	48
4.3.1	Armazenamento de dados	48
4.3.2	Publicação de dados	52
4.3.3	Acesso aos dados	53
4.3.4	Serviços	54
4.3.5	Camada de interação com o utilizador	58
4.4	Desenvolvimento	59
4.4.1	Pesquisas.....	59
4.4.2	Camada de negócio	62
4.4.3	Interface gráfica.....	63
4.5	Avaliação.....	70
4.5.1	Quão amigável classifica a interface da aplicação?	70
4.5.2	Está satisfeito com as funcionalidades disponibilizadas?.....	70
4.5.3	A informação disponibilizada é útil no seu dia-a-dia?	70
4.5.4	Acha as novas funcionalidades úteis para a melhoria do desempenho do seu armazém?	71
4.5.5	Como responsável de armazém, com que frequência espera utilizar as novas funcionalidades?	71
4.5.6	Sugestões	71
4.6	Conclusões	71
5	Conclusão	73
5.1	Resumo do trabalho realizado	73
5.2	Trabalho futuro	74
6	Referências	75
A1.	Inquérito de satisfação.....	77

Lista de Figuras

Figura 1: Ponto, linha e polígono	16
Figura 2: Representação cartesiana e esférica	17
Figura 3: Distribuição global de vegetação	17
Figura 4: Planta de habitação.....	18
Figura 5: Representação de um polígono e dos pontos utilizados para sua definição.....	20
Figura 6: Modelo de dados	24
Figura 7: Diagrama de Entidade e Relacionamento.....	25
Figura 8: Mapeamentos Elasticsearch	26
Figura 9: Corpo do pedido para a rota entre duas cidades.....	29
Figura 10: Simulação de viagem.....	29
Figura 11: Diagrama de implementação	42
Figura 12: Casos de uso.....	45
Figura 13: Diagrama de atividade - ver motoristas.....	46
Figura 14: Diagrama de atividade - motoristas mais próximos	47
Figura 15: Diagrama de atividades - pesquisa de motoristas	48
Figura 16: Modelo de dados	49
Figura 17: Elasticsearch <i>schema</i>	50
Figura 18: Diagrama de classes – negócio	51
Figura 19: Diagrama de classes - UI	52
Figura 20: Fluxo de dados no RabbitMQ.....	53
Figura 21: Diagrama de Sequência - Localização de motoristas.....	55
Figura 22: Diagrama de Sequência - Encontrar motorista	56
Figura 23: Diagrama de Sequência - Motoristas mais próximos.....	57
Figura 24: Organização da interface gráfica.....	59
Figura 25: Pesquisa - motoristas mais próximos.....	61
Figura 26: Classe para definição dos pedidos	62
Figura 27: Localização de motoristas	65
Figura 28: Encontrar motorista	67
Figura 29: Motoristas mais próximos.....	69

Lista de Tabelas

Tabela 1: <i>Software</i> desenvolvido na Jumia Services.....	2
Tabela 2: Casos de teste de leitura de informação.....	5
Tabela 3: Proposta de valor - Benefícios e Sacrifícios.....	8
Tabela 4: Proposta de valor - Benefícios e Sacrifícios.....	9
Tabela 5: Modelo de negócio de CANVAS	9
Tabela 6: Categorias de bases de dados não relacionais.....	13
Tabela 7: PostGIS - representação de informação.....	15
Tabela 8: Subtipos de Ponto	19
Tabela 9: Subtipos de Linha	19
Tabela 10: Funções do PostGIS	20
Tabela 11: Operadores.....	21
Tabela 12: Funções disponibilizadas pelo Elasticsearch.....	22
Tabela 13: Cidades mais populadas de Portugal.....	27
Tabela 14: Infraestrutura aplicacional	43
Tabela 15: Tecnologias utilizadas.....	44
Tabela 16: Search Templates	53
Tabela 17: Serviços disponibilizados pelo HMT	54
Tabela 18: Códigos de resposta HTTP utilizados.....	63

Lista de gráficos

Gráfico 1: Registo de dados	30
Gráfico 2: Teste de escrita – Base de dados com 1 000 000 de registos	31
Gráfico 3: Inserção em bulk	31
Gráfico 4: Inserção em bulk. Base de dados com 1 000 000 de registos.....	32
Gráfico 5: Média do tempo consumido na inserção de dados	33
Gráfico 6: Total de registos na tabela de localizações	34
Gráfico 7: Obtenção de 200 localizações	34
Gráfico 8: Número de localizações por motorista	35
Gráfico 9:Leitura da última localização de cada um dos motoristas	35
Gráfico 10: Localização atual dos motoristas e últimos 10 pontos.....	36
Gráfico 11: Localização de motoristas dentro de um retângulo.....	36
Gráfico 12: Localização atual dos motoristas dentro de um quadrado e últimos 10 pontos	37
Gráfico 13: Localização de motoristas dentro de uma circunferência	37
Gráfico 14: Localização de motoristas dentro de uma circunferência e últimos 10 pontos	38
Gráfico 15: Consumo de memória pelas bases de dados	39

Acrónimos e Símbolos

Lista de Acrónimos

AIG	<i>Africa Internet Group</i>
API	<i>Application Programing Interface</i>
ETL	<i>Extract Transform and Load</i>
GPS	<i>Global Positioning System</i>
JSON	<i>JavaScript Object Notation</i>
MBR	<i>Minimum Bounding Rectangle</i>
SGBD	<i>Sistema de Gestão de Bases de Dados</i>
RDBMS	<i>Relational database manahement system</i>
SQL	<i>Structured Query Language</i>

1 Introdução

O presente trabalho surge no âmbito da unidade curricular de Tese / Dissertação do curso de Mestrado em Engenharia Informática, ramo de Sistemas Gráficos e Multimédia, do Instituto Superior de Engenharia do Porto. Esta tese foi desenvolvida em ambiente empresarial, no Porto Tech Center do Grupo Africa Internet Group.

1.1 Contexto

Desde a sua criação, em 2012, a Africa Internet Group (AIG) já criou 71 empresas em diferentes setores, e.g. plataforma de compra de alimentos, mercado *online*, agência de viagens online, mercado imobiliário, entre outros.

O comércio *online* é uma tendência que já está estabelecida na maior parte do hemisfério ocidental (Africa Internet Group, 2016). O mercado *online* atingiu, em 2013, vendas superiores 1.2 mil milhões de dólares, sendo cerca de 5% deste valor, obtido no Médio Oriente e África. A mesma fonte prevê uma evolução de 100% até 2018 (“B2C e-commerce sales in MEA 2017 | Statistic,” n.d.).

Inerente à evolução das plataformas de venda *online* de produtos está o aparecimento de empresas que se dedicam à distribuição desses produtos. É neste setor que surge Jumia Services, uma nova empresa do grupo AIG, que pretende ter a maior rede de logística no continente Africano. Ativa em 12 países, para além dos vasto leque de serviços de logística prestados pela empresa, encontra-se a desenvolver soluções de *software* para dar suporte à prestação desses mesmos serviços.

No que toca aos serviços de logística, alguns exemplos são:

- Entrega de bens;
- Recolha de bens;
- Recolha de bens do fornecedor e entrega no cliente;
- Devolução de bens do cliente para o fornecedor;
- Entre outros.

No que toca ao desenvolvimento de *software*, a Tabela 1 apresenta o catálogo de produtos a serem desenvolvidos, alguns deles já com versão em produção, juntamente com uma breve descrição da sua função.

Tabela 1: *Software* desenvolvido na Jumia Services

Software	Descrição
API	Este módulo é a interface da Jumia Services com os seus clientes. Os clientes enviam os pedidos para esta interface que os processam de forma assíncrona. As atualizações dos estados dos pedidos são enviadas de volta para os clientes.
RMS	O módulo de gestão de pedidos (RMS – Request Management System) é responsável por redirecionar os pedidos dos clientes para os módulos respetivos. Por exemplo, se se trata de um pedido a um armazém, o RMS irá escolher o armazém mais indicado para processar o pedido (com base na localização, stock, etc.) e depois redirecionar o pedido para o WMT.
HMT	O HMT (<i>Hub Management Tool</i>) é um <i>software</i> utilizado por todas entidades envolvidas no funcionamento do armazém. Desde o responsável de armazém, que necessita de ter uma visão clara das tarefas que têm de ser realizadas, até ao colaborador que faz a receção dos pacotes, em que um leitor de códigos de barras é tudo o que precisa para exercer as suas funções, o HMT disponibiliza uma solução intuitiva que requer pouca formação. O principal objetivo do HMT é conduzir o utilizador à sua próxima tarefa e assegurar que esta é realizada corretamente.
WMT	<i>Warehouse Management Tool</i> – É uma solução equivalente ao HMT mas direcionado para os armazéns. O tipo de tarefas executadas num armazém são muito específicas: controlo de stock, gestão de localização do stock.
D-app	Durante o seu dia-a-dia, os motoristas da Jumia Services fazem-se acompanhar de um dispositivo móvel através do qual têm acesso a esta aplicação mobile. Através da aplicação, podem registar a entrega de uma encomenda, identificar o próximo local de entrega, entre outros.
Live reporting	As operações necessitam de relatórios para tomar as decisões acertadas. A equipa de BI (<i>Business Intelligence</i>) produz vários relatórios que incluem a análise de dados de dias/semanas anteriores. O módulo de representação de informação em tempo real está focado no que está a acontecer no presente, o que permite retirar uma análise das tendências ao longo do dia. Além disto, inclui também a localização e estado atual das encomendas. Assim, o objetivo deste sistema é fornecer os dados muito rapidamente.

De entre as soluções apresentadas na tabela, apenas uma é utilizada durante o processo de entrega e recolha de pacotes. A D-app pretende ser uma aplicação que acompanha os motoristas durante o seu dia-a-dia para registar todo o processo de entrega de bens.

O surgimento da D-app permite que a atualização do estado dos bens processados pela Jumia Services ocorra em tempo real, na medida em que os motoristas utilizam esta aplicação para

registrar o sucesso/insucesso da entrega/recolha de um bem em tempo útil. Para além da atualização do estado dos bens, a utilização ativa da aplicação permite que a informação da localização geográfica dos motoristas seja partilhada com outros sistemas.

1.2 Problema

A massificação dos recetores de GPS trouxe um aumento da informação disponível, aumentando a relevância das bases de dados espaciais. Da tradicional pesquisa geoespacial em bases de dados, passamos agora para uma necessidade de avaliação, em tempo real, da informação. Por exemplo, num sistema com o *tracking* periódico das coordenadas GPS, interessa saber se uma das máquinas está a tomar uma rota não desejada ou longe do ótimo por alguma definição de ótimo que considere, por exemplo, a distância e o custo.

Adicionalmente, a evolução dos motores de busca em direção a bases de dados não relacionais levanta a questão adicional de como integrar a informação espacial neste sistema.

O acesso a informação em tempo real pode ser crucial para a gestão eficiente dos recursos de uma empresa. No caso particular da Jumia Services, o acesso em tempo real à dos seus motoristas poder-se-á tornar útil na:

- Identificação dos motoristas mais próximos de locais onde é necessário proceder à recolha de um bem. Com isto, será possível diminuir o tempo de demora na execução de um serviço;
- O registo do percurso realizado por motorista permitirá construir um universo de dados de onde se poderá extrair informação relativa ao desempenho do mesmo.

As soluções de *software* existentes atualmente na empresa não dão resposta a esta necessidade.

Uma outra questão que se levanta aquando o desenho e conceção deste tipo de soluções é a da tecnologia mais adequada para fim pretendido. Por exemplo, em aplicações com o WMT e o HMT (descritos na Tabela 1) são utilizadas bases de dados relacionais que garantem a integridade dos dados, enquanto que no caso do Dashboard (*Live reporting* descrito na Tabela 1) torna-se mais importante o acesso eficiente informação. Assim, ainda antes de desenvolver um produto que disponibilize, em tempo real, o acesso à localização dos motoristas, torna-se necessário a identificação das tecnologias mais indicadas para o desenvolvimento do mesmo.

A fonte de informação, no que toca à localização dos motoristas em tempo real, é a aplicação mobile, que faz regularmente a publicação da localização geográfica dos motoristas. A frequência de registo de dados está dependente do modo em que o utilizador se encontra:

- *Low mode* – a cada 5 minutos;
- *Run mode* – a cada 30 segundos;
- *Emergency mode* – a cada 5 segundos.

Através desta aplicação, os colaboradores da Jumia Services fazem também o registo de vários eventos como por exemplo:

- Início de entrega de um pacote;
- Pacote entregue com sucesso;
- Tentativa de entrega falhada;

Assim, é com base nos dados disponibilizados por esta aplicação que se pretende que a monitorização dos motoristas seja realizada.

1.3 Objetivo e prova de conceito

Esta secção pretende apresentar os objetivos delineados para o projeto descrito neste relatório. Pretende-se assim responder ao problema identificado na secção 1.2.

O objetivo principal para a realização deste projeto é a avaliação do desempenho de bases de dados, conforme descrito na secção 1.3.1. Contudo, o desenvolvimento de um protótipo que permita ter acesso, através da criação de um novo módulo numa das aplicações da suite de produtos da Jumia Services, à localização dos motoristas da empresa (descrito na secção 1.3.2) serve como prova de conceito e pretende mostrar a viabilidade de um componente de rastreio de motoristas baseado no estudo efetuado.

1.3.1 Avaliação de desempenho de bases de dados

Escolher o sistema de armazenamento e pesquisa de informação é um passo importante na implementação de um sistema de disponibilização de dados em tempo real. É neste sentido que surge o objetivo do presente projeto, que passa pela análise de desempenho de duas diferentes soluções de armazenamento e pesquisa de informação (ElasticSearch e PostgreSQL). A escolha destas duas soluções para o armazenamento e pesquisa de informação assenta no fato de serem as tecnologias já adotadas na empresa nos seus produtos.

No conjunto de projetos de *software* desenvolvidos na Jumia Services são utilizadas diversas tecnologias, desde a camada de interface com os utilizadores até aos sistemas de armazenamento e pesquisa de informação. Para o armazenamento de informação, grande parte dos produtos utilizam a solução relacional do PostgreSQL. Contudo, para as funcionalidades que requerem acesso eficiente aos dados como as pesquisas de texto livre ou o cálculo de métricas de desempenho, é utilizada a solução não relacional ElasticSearch.

No contexto da localização de motoristas, pretende-se saber qual das duas bases de dados (PostgreSQL e ElasticSearch) apresenta um melhor desempenho quando testada em vários cenários. Pretende-se assim que os cenários utilizados abranjam os seguintes tópicos:

- Utilização de espaço, em disco, das diferentes bases de dados com o mesmo número de registos;

- Velocidade de escrita de registos singulares. Isto é, medir o tempo utilizado por cada uma das bases de dados para proceder à escrita de N registos, 1 de cada vez;
- Velocidade de escrita de registos em massa. Num universo de dados com N registos, pretende-se que esses registos sejam divididos em diferentes grupos e que sejam inseridos cada um dos grupos de uma só vez na base de dados. O objetivo aqui é comparar os tempos de escrita;
- Velocidade de leitura de informação nas diferentes bases de dados;

Os pontos identificados na lista acima assinalam os aspetos a ter em conta no processo de análise de desempenho das bases de dados. No conjunto identificam os parâmetros considerados de maior importância para a escolha do sistema de armazenamento de dados da aplicação desenvolvida na segunda fase do projeto aqui descrito. Estes parâmetros são a velocidade de leitura e de escrita nas bases de dados.

Neste sentido, na Tabela 2 encontram-se inumerados os casos de teste utilizados para a avaliação, que vão ao encontro das necessidades do *software* a ser desenvolvido na segunda fase deste projeto.

Tabela 2: Casos de teste de leitura de informação

Identificador	Descrição
Escrita	
E.1	Escrita de 100 000 localizações, em blocos de 10 000, de motoristas numa base de dados vazia.
E.2	Escrita de 100 000 localizações, em blocos de 10 000, de motoristas numa base de dados com 1 000 000 de registos.
E.3	Escrita de 100 000 localizações, em blocos de 10 000, com recurso às operações de <i>bulk</i> disponibilizadas por ambas as bases de dados.
E.4	Escrita de 100 000 localizações, em blocos de 10 000, de motoristas numa base de dados com 1 000 000 de registos. Esta escrita utiliza as operações de <i>bulk</i> disponibilizadas pelas bases de dados.
Leitura	
L.01	Leitura da localização atual de todos os motoristas.
L.02	Leitura dos últimos 10 pontos geográficos de cada um dos motoristas.
L.11	Leitura de todos os motoristas que se encontram dentro de uma área pré definida.
L.12	Leitura dos últimos 10 pontos de todos os motoristas que se encontram dentro de uma área pré definida.
L.21	Leitura da localização de todos os motoristas dentro de uma circunferência.
L.31	Leitura de todos os motoristas que se encontram dentro de uma circunferência maior do que a realizada no teste anterior.

1.3.2 Desenvolvimento de aplicação protótipo

No seguimento da análise de desempenho das bases de dados foi proposto o desenvolvimento de uma prova de conceito que disponibiliza um conjunto básico de funcionalidades no contexto de *tracking* de motoristas.

O objetivo desta prova de conceito passa por compreender, do ponto de vista do utilizador final da aplicação, se é possível construir uma solução satisfatória com o desempenho da Base de Dados adotada para a solução. Tratando-se de uma prova de conceito, deverá ser desenvolvida apenas uma versão com a utilização da Base de Dados considerada como melhor solução, com base no estudo realizado na primeira fase do projeto.

Com o desenvolvimento deste protótipo será possível determinar o nível de satisfação daqueles que serão os utilizadores finais de uma solução de *tracking*, a ser desenvolvida posteriormente na Jumia Services.

Sendo este um objetivo proposto pelo autor do projeto, será realizado um inquérito de satisfação aos utilizadores que permita compreender dois aspetos:

- Grau de satisfação com a usabilidade da aplicação. Com isto pretende-se saber se os utilizadores consideram as novas funcionalidades fáceis de utilizar. Este é um *feedback* importante na medida em que a aplicação se destina a colaboradores de países com culturas diferentes das culturas Europeias e pelo fato de estes apresentarem graus de escolaridade mais baixos;
- Compreender a frequência com que os utilizadores pretendem utilizar a aplicação no seu dia-a-dia e se as funcionalidades disponibilizadas lhes pode ser útil na melhoria do seu desempenho.

A importância perceber a satisfação do utilizador no que toca à interface da aplicação prende-se com o facto de esta ser utilizada por um público que possui diferentes níveis de contato com a tecnologia, o que pode dificultar o modo com que interagem com a aplicação. No que toca às questões relativas à informação prestada pelas novas funcionalidades e a utilidade que as mesmas têm para os utilizadores, a sua importância deve-se à necessidade de ter uma perspetiva de se este é o caminho a seguir no que toca ao desenvolvimento de uma solução de *tracking* de motoristas.

No sentido de obter respostas sinceras, o questionário será realizado de forma anónima.

Para a implementação deste protótipo foi escolhido o HMT (descrito na Tabela 1) pelo fato de ser a aplicação utilizada nos armazéns da empresa de logística e pela possibilidade de levar até aos utilizadores, no terreno, as novas funcionalidades desenvolvidas.

Assim, com a implementação deste protótipo propôs-se o desenvolvimento das funcionalidades apresentadas nas secções que se seguem (1.3.2.1, 1.3.2.2 e 1.3.2.3).

1.3.2.1 Localização de motoristas

A primeira funcionalidade passa por identificar todos os parceiros ativos, nos últimos 5 minutos, numa área definida pelo utilizador.

O utilizador da aplicação deverá ter a possibilidade de definir, num mapa, uma área dentro da qual serão identificados todos os motoristas. Os parceiros deverão ser identificados no mapa sob a forma de círculos e a informação atualizada a cada 5 segundos.

A par com a apresentação visual da localização dos motoristas, deverá ser apresentada a lista com informação detalhada de cada um deles. Através da seleção de um motorista, a aplicação disponibiliza ao utilizador a possibilidade de visualizar, no mapa, apenas o motorista selecionado assim como a rota que o mesmo se encontra a percorrer.

1.3.2.2 Identificação dos motoristas mais próximos

Através da seleção de um ponto no mapa e de um raio máximo, a aplicação deverá disponibilizar, ao utilizador, a possibilidade de encontrar os motoristas mais próximos.

1.3.2.3 Encontrar motorista

Atendendo à possibilidade de que um motorista pode não se encontrar ativo no momento em que o utilizador se encontra a utilizar o sistema, a aplicação deverá disponibilizar uma funcionalidade de pesquisa de parceiros.

Utilizando o nome, ou parte, assim como outro elemento identificador do motorista, o utilizador deverá ser capaz de encontrar todos os motoristas cujos dados combinem com o texto inserido. Selecionando um dos resultados da pesquisa, a aplicação deverá conduzir o utilizador até à página de detalhe do parceiro.

1.4 Abordagem metodológica

O trabalho aqui apresentado foi constituído por três fases principais. Numa primeira fase foram estudados os conceitos e tecnologias relacionados com o projeto proposto. Foram assim estudadas as soluções de armazenamento de dados espaciais disponibilizados pelas bases de dados atualmente em uso na Jumia Services. A segunda fase passou pela realização de um estudo comparativo das bases de dados utilizadas na Jumia Services no contexto do desenvolvimento de uma solução de *tracking* de motoristas em tempo real.

Por fim, considerando os resultados obtidos na análise comparativa das bases de dados e atendendo aos aspetos de interesse desse mesmo estudo, foi desenvolvido um protótipo de localização em tempo real dos motoristas da Jumia Services. Este protótipo utiliza a Base de Dados que, depois de realizados os testes de desempenho, foi definida como a melhor solução para o armazenamento e pesquisa de informação.

1.5 Valor de negócio

A proposta de valor constitui uma peça principal num modelo de negócio. Para a definição de um modelo de negócio é necessário uma visão clara e focada da proposta de valor. Sem esta não é possível determinar se e como será gerada receita, os parceiros necessários e como serão obtidos e mantidos os clientes. (*Crafting your value proposition*, 2012) De uma forma sucinta, o mesmo autor cita um outro para definir a proposta de valor como sendo uma declaração dos benefícios funcionais, emocionais e auto expressivos entregues por uma marca, que constituem valor para o cliente final.(Aaker, 1996)

A proposta de valor deve ser clara ao ponto de facilmente descrever o produto em causa, qual o seu público-alvo, quais as necessidades desse mesmo público e, por fim, de que forma o produto irá resolver os problemas desse mesmo público.

No processo de criação de um negócio, é necessário ainda ter em consideração o valor percebido pelo cliente, que constitui a avaliação total realizada pelo consumidor relativamente à utilidade do produto, baseada em perceções do que é recebido (benefícios) e do que é dado (sacrifícios) (Zeithaml, 1988).

O projeto descrito neste documento pretende criar valor, a nível técnico e de negócio, uma vez que está dividido em duas componentes bastantes distintas mas relacionadas entre si.

O desenvolvimento de um estudo comparativo do desempenho de diferentes tipos de Bases de Dados permitirá ter uma noção clara do comportamento das mesmas perante situações idênticas. Assim, numa fase inicial de desenvolvimento de um projeto, em que é necessário optar pela utilização de determinadas tecnologias em detrimento de outras, este estudo revela-se uma importante ferramenta com o fornecimento de dados relativos ao desempenho de duas soluções em concreto.

A Tabela 3 apresenta os benefícios/ sacrifícios, numa perspetiva longitudinal de valor, relativamente a esta primeira fase do projeto.

Tabela 3: Proposta de valor - Benefícios e Sacrifícios

	Serviço	Relacionamento
Benefícios	Customização Preço	Conflito
Sacrifícios		Tempo Conflito

Conforme apresentado na Tabela 3, o preço surge como um benefício do serviço na medida em que o estudo realizado é feito de forma gratuita. Também a customização é apresentada como um benefício uma vez que o estudo é realizado tendo por base as necessidades da empresa onde o mesmo é realizado, considerando as tecnologias utilizadas pela empresa para efeitos comparativos. O Conflito é classificado com um benefício sendo que não foram encontrados estudos que relacionassem Bases de Dados relacionais e não relacionais. Contudo, o mesmo é

apresentado como um sacrifício porque estão disponíveis estudos que comparam um maior número de soluções.

Por fim, o Tempo surge como um sacrifício tendo sido considerado o tempo necessário à especificação dos múltiplos casos de teste, o tempo necessário ao desenvolvimento dos mesmos e à identificação das conclusões resultantes do estudo.

O desenvolvimento de uma prova de conceito pretende disponibilizar uma solução de *tracking* de parceiros. Nesta medida, surge como uma ferramenta de apoio à decisão uma vez que disponibiliza a localização dos motoristas dos armazéns da empresa. Na perspetiva de evolução do produto, esta aplicação poderá garantir o acesso aos dados em tempo real, passar a contabilizar *KPI's* que permitam categorizar os parceiros numa escala de qualidade do serviço prestado e disponibilizar, para os consumidores da Jumia Services, uma solução de *tracking* das suas encomendas.

É ainda possível ver este produto numa perspetiva de apoio à tomada de decisão. Por exemplo, através do conhecimento da localização atual dos motoristas, na eminência do aparecimento de um novo serviço de recolha, é possível selecionar o parceiro que se encontra mais próximo do ponto de recolha e seleciona-lo para a realização do novo serviço.

No seguimento do desenvolvimento do protótipo descrito, a Tabela 4 apresenta os benefícios e sacrifícios numa perspetiva longitudinal de valor.

Tabela 4: Proposta de valor - Benefícios e Sacrifícios

	Serviço	Relacionamento
Benefícios	Customização Preço	
Sacrifícios		Tempo Conflito

1.5.1 Modelo de negócio de CANVAS

O modelo de negócio de CANVAS constitui uma ferramenta de planeamento estratégico que permite desenvolver modelos de negócio. No sentido de definir os fatores associados ao negócio, a Tabela 5 apresenta o modelo referido.

Tabela 5: Modelo de negócio de CANVAS

Parceiros chave	Atividades principais	Proposição de valor	Relação com os clientes	Segmento de clientes
O principal parceiro será a AIG, empresa onde o projeto será desenvolvido e onde será instalado o projeto piloto.	Desenvolvimento de <i>software</i> de localização, em tempo real. Principais recursos Servidores;	Otimização de custos em serviços de entrega; Utilização de tecnologias <i>open source</i> ; Rastreamento, em tempo real, da localização da frota.	Ajuste do <i>software</i> às necessidades do cliente. Canais Através do principal parceiro – AIG, uma	Empresas de logística; Empresas com necessidade de localização de dispositivos em tempo real;

	Equipa de desenvolvimento: Ferramentas de desenvolvimento.	Estudo comparativo de bases de dados relacionais e não relacionais.	vez que possui <i>software</i> de gestão de logística.	
Estrutura de custos Equipa de desenvolvimento; Servidores; Infraestrutura de tecnologias e informação;		Fluxos de receita As receitas serão alcançadas através da venda do produto		

1.6 Estrutura da tese

Este documento encontra-se dividida em vários capítulos, de forma a melhor descrever o desenrolar do projeto. O capítulo 1 faz uma introdução ao tema abordado deste trabalho de mestrado, fazendo o enquadramento do projeto bem como a apresentação da organização onde o mesmo foi realizado. São ainda apresentadas as motivações que levaram à realização deste trabalho.

No capítulo 2 é feita uma revisão bibliográfica que tem como objetivo a abordagem dos conceitos necessários ao entendimento dos assuntos aqui abordados.

O desenvolvimento do projeto encontra-se descrito em dois capítulos distintos (capítulos 3 e 4). A análise ao desempenho realizada às Bases de Dados encontra-se descrita na secção 3 enquanto que a secção 4 faz a descrição técnica da prova de conceito desenvolvida. Cada um destes capítulos apresenta as conclusões parciais do trabalho desenvolvido.

O capítulo 5 apresenta as conclusões gerais do projeto descrito neste relatório.

Por fim, no capítulo 6 são apresentadas as referências bibliográficas utilizadas para o desenvolvimento do projeto.

2 Revisão bibliográfica

A forma mais comum de armazenamento de dados nos dias de hoje é com recurso ao modelo relacional. Contudo, as soluções não relacionais têm vindo a ganhar destaque uma vez que permitem o tratamento de grandes quantidades de informação com elevado desempenho.

Este capítulo faz uma abordagem ao conceito de bases de dados, caracterizando as relacionais e as não relacionais. São ainda aqui apresentados alguns estudos que comparam o desempenho destes dois tipos de sistemas de armazenamento de dados.

2.1 Considerações gerais

As Bases de Dados são sistemas responsáveis pelo armazenamento de informação digital, de forma que seja possível manipulação e consulta da mesma. A evolução das aplicações web trouxe a necessidade do processamento de um número crescente de informação.

A opção por um Sistema de Gestão de Bases de Dados (SGBD) recai, não só pela garantia de persistência dos dados a armazenar, como também pela integridade dos mesmos.

O crescimento exponencial da quantidade de informação armazenada nas bases de dados constitui um problema para muitas empresas. Com vários anos de dados armazenados, estes sistemas precisam de continuar a demonstrar eficiência no que diz respeito ao processamento e ao armazenamento dos dados. Esta necessidade de aceder de forma eficiente aos dados de uma aplicação conduziu ao aparecimento de alternativas às tradicionais Bases de Dados relacionais.

2.1.1 Bases de Dados relacionais

As Bases de Dados relacionais caracterizam-se pelo modelo relacional, armazenando os dados em linhas e colunas de tabelas de forma estruturada. Associado ao conceito de Bases de Dados surge o de Sistemas de Gestão de bases de dados (SGBD), responsável pela forma como os dados são armazenados, mantidos e devolvidos para as aplicações cliente. Ainda no contexto

deste tipo de Bases de Dados, existe uma linguagem para a realização de pesquisas e manutenção dos dados, SQL – *Structured Query Language*.

Os SGBD têm atualmente amplo suporte, com existência de ferramentas para a gestão de utilizadores e permissões, cópias de segurança e recuperação de dados, e os métodos para assegurar a integridade dos dados. Os SGBD são utilizados para o suporte de inúmeras aplicações tendo este mercado, segundo um relatório do IDC (*International Data Corporation*), atingido receitas na ordem dos 26 mil milhões de dólares (em 2011) e estando previsto alcançar os 41 mil milhões em 2016. (Gudivada, et al., 2014)

Os dados neste tipo de sistemas encontram-se tipicamente normalizados, o que resulta na criação de inúmeras tabelas (dependendo da dimensão do negócio representado) e relações entre as mesmas. As pesquisas com este tipo de estruturas requer a análise de diferentes tabelas com a realização de combinações entre elas. Quanto maior for o número de tabelas envolvidas, mais complexa será a realização de pesquisas, o que resultará numa maior demora para a obtenção dos dados pretendidos.

As transações num sistema relacional trazem inúmeras vantagens, como por exemplo, a garantia da integridade dos dados que são persistidos na mesma. Contudo, este conceito traz também um certo peso às ações que são feitas na mesma, uma vez que as transações são caracterizadas pelas seguintes propriedades.

- **Atomicidade:** Quando uma base de dados processa uma transação, garante que ou a transação é concluída com sucesso ou todos os passos anteriores ao ponto de falha são anulados. Por exemplo, numa transação com 5 passos, caso ocorra um erro no processamento do terceiro passo, serão revertidas as operações realizadas no primeiro e segundo passos.
- **Consistência:** Uma vez definido um modelo de dados, com regras para a integridade dos mesmos, como é o caso das *constraints*, o SGBD não permite a persistência de dados o que não estejam em conformidade com essas regras.
- **Isolamento:** Esta característica nos SGBD garante que, transações que ocorram em simultâneo, não tenham impacto na execução de cada uma delas.
- **Durabilidade:** As transações que ocorrem com sucesso devem ser mantidas no sistema. Antes de terminar a execução da transação, estas alterações devem ficar registadas no histórico de transações. Na eventualidade de ocorrer uma falha no sistema, todas as transações que não foram concluídas com sucesso, devem ser processadas novamente.

A gestão de transações num SGBD pode ser visto como uma vantagem no que toca à integridade dos dados. No entanto, pode também ser visto como uma desvantagem em termos de desempenho das operações realizadas na base de dados.

Uma outra questão relacionada, não só com as Bases de Dados, mas com as soluções de *software* em geral, é a capacidade destes sistemas escalarem – escalabilidade. No caso específico das Bases de Dados, esta representa a capacidade de suportar a evolução do esquema de dados.

As vantagens dos SGBD, aqui mencionadas, fazem destes sistemas ótimas alternativas no armazenamento de dados para muitas aplicações. Contudo, isto não significa que representem a melhor solução para todas as aplicações.

2.1.1.1 PostgreSQL

Como mencionado anteriormente, o projeto descrito neste documento pretende realizar um estudo comparativo de desempenho das Bases de Dados relacionais e não relacionais. O SGBD utilizado para este estudo foi o PostgreSQL, uma vez que é a solução de armazenamento de dados adotada pela empresa onde este projeto foi desenvolvido.

No que toca à representação de dados, assim como outros SGBD, o PostgreSQL apresenta a sua informação representada sob a forma de tabelas, linhas e relações. A definição do esquema de Bases de Dados (definição de tabelas) é um passo obrigatório para o armazenamento da informação uma vez que não é possível proceder ao armazenamento de informação sem que as tabelas estejam definidas. Todos os registos de uma tabela têm, obrigatoriamente, de ter o mesmo número de colunas.

A normalização do esquema de dados no PostgreSQL é feita através da separação da informação em tabelas com a definição de relações entre as mesmas. Desta forma, reduz-se a redundância dos dados, evitando que os mesmos se tornem inconsistentes ao longo do tempo.

O suporte a transações é outra importante característica importante do PostgreSQL uma vez que permite a definição de um conjunto de operações como um único passo a ser processado pelo SGBD.

2.1.2 Bases de dados não relacionais

Segundo a definição apresentada por Okman, Gal-Oz, Gonen, Gudes e Abramov em 2011, as Bases de Dados não relacionais são altamente escaláveis, confiáveis, com um modelo de dados simples, sem mecanismo para lidar com a consistência e integridade dos dados e quase sem suporte para a segurança a nível de Bases de Dados.

Dependendo da Base de Dados escolhida, existem diferentes modos de armazenamento de informação. Enquanto as Bases de Dados relacionais utilizam um sistema relacional, as bases de dados *NoSQL* fazem-no de forma dependente da categoria. A Tabela 6 apresenta 3 exemplos de categorias de Bases de Dados não relacionais.

Tabela 6: Categorias de bases de dados não relacionais

Categoria	Descrição
<i>Key Value</i>	Tal como o nome sugere, neste modelo os valores estão associados a uma chave. A estrutura desta categoria é a mais simples. A chave é uma <i>String</i> e o valor um <i>BLOB</i> . A flexibilidade deste modelo permite aos programadores associar os valores a objetos de programação (no formato JSON, por exemplo).

<i>Document Database</i>	Constitui uma extensão à categoria Key Value. Os documentos podem ser armazenados em vários formatos binários (por exemplo, PDF) ou no formato ASCII (por exemplo, XML, JSON). Exemplos deste tipo de NoSQL DB são o ElasticSearch, Mongo DB e Couch DB.
<i>Graph database</i>	Este tipo de Bases de Dados caracteriza-se pela eficiência no armazenamento e consulta de dados ricos em relacionamento, como é o caso dos grafos. Este tipo de BD usa nós, relações e propriedades para o armazenamento de dados. Em comparação com as Bases de Dados relacionais, os nós são tabelas, as relações entre os nós assemelham-se às relações entre as tabelas e as propriedades são colunas das tabelas. Exemplo deste tipo de NoSQL BD é o Neo4j.

Ao contrário dos SGBD, as Bases de Dados não relacionais não estão construídos de acordo com o ACID (Atomicidade, consistência, integridade e durabilidade). Esta característica pode trazer vantagens no que toca ao desempenho das operações realizadas no sistema contudo, torna-se um sistema mais frágil em termos de integridade e consistência da informação.

2.1.2.1 *ElasticSearch*

O ElasticSearch foi a Bases de Dados não relacional utilizada para a realização deste estudo, uma vez que, de forma semelhante ao PostgreSQL, é um sistema já utilizado na empresa onde a dissertação foi desenvolvida.

O ElasticSearch é uma Bases de Dados escalável horizontalmente. Desta forma, permite o tratamento de uma grande quantidade de informação. Este é um sistema distribuído por múltiplas máquinas, o que aumenta a sua tolerância a falhas.

No que toca à representação dos dados, o ElasticSearch é uma *document database*. Apesar de ser uma Bases de Dados não relacional, permite manter um tipo de relações entre os documentos (relação *parent-child*). Para melhor compreender este sistema de armazenamento de informação, podemos considerar que um índice, no ElasticSearch, representa a mesma importância de uma tabela numa base de dados relacional.

Cada um dos índices tem um mapeamento associado, que define os seus campos, como são definidas as colunas nas tabelas de uma base de dados relacional. Contudo, na eventualidade de ser introduzido um documento com um número diferente de campos, o índice ajustar-se-á e o seu mapeamento será alterado para contar com a presença dos novos campos.

De forma a possibilitar a consulta de dados, o ElasticSearch disponibiliza uma linguagem, no formato JSON, denominada *Query DSL*. As pesquisas podem ser feitas sob a forma de *queries* ou filtros, sendo que a diferença reside no fato de a *query* associar uma pontuação a cada documento enquanto que o filtro não o faz. Isto faz com que os filtros tornem a pesquisa mais rápida.

Uma importante limitação do Elasticsearch é o fato de não permitir que seja alterado o mapeamento de um índice. Ao contrário das Bases de Dados relacionais, em que é possível alterar a estrutura de uma tabela, acrescentando ou reduzindo o número de colunas, para adicionar um novo campo a um mapeamento, é necessário proceder à reindexação de todos os dados presentes nesse índice.

2.2 PostgreSQL e PostGIS

Um Sistema de Gestão de Bases de Dados Relacional (RDBMS – *Relational Database Management System*), referido de forma mais comum como Base de Dados, é uma complexa suíte de aplicações de *software* que controlam a organização, armazenamento e acesso à informação. As Bases de Dados disponíveis suportam um vasto leque de tipos de dados. O sistema de tipos de dados pode ser alargado com vista a oferecer não só um maior número de tipos mas também o número de operações suportadas. (Power, 2009)

O termo Base de Dados espacial descreve um RDBMS que suporta informação geográfica da mesma forma que suporta outro tipo de dados. Este conjunto de tipos de dados pode incluir:

- Ponto;
- Linha;
- Polígono.

O PostgreSQL não providencia, por omissão, o suporte para o armazenamento, processamento e indexação de dados espaciais. Para isso, disponibiliza uma extensão PostGIS. As extensões espaciais, para além da oferta em termos de tipos de dados e funções, providenciam também novos algoritmos de indexação de informação no sentido de tornar o acesso a este tipo de dados mais eficiente.

As próximas secções deste documento (2.2.1 a 2.2.3) abordam, com mais detalhe, as características da extensão PostGIS.

2.2.1 Representação de dados

A extensão PostGIS disponibiliza quatro diferentes representações de tipos espaciais: *geometry*, *geography*, *raster* e *topology*. A Tabela 7 apresenta uma breve descrição das representações mencionadas, sendo as mesmas apresentadas com mais detalhe nas secções 2.2.1.1 a 2.2.1.4.

Tabela 7: PostGIS - representação de informação

Tipo	Descrição
<i>Geometry</i>	Caracteriza-se por ser um tipo plano. Foi o primeiro tipo espacial disponibilizado e mantém-se o mais utilizado de entre os suportados.
<i>Geography</i>	Tipo geodésico esferoidal. Disponibiliza o desenho de linhas e polígonos na superfície curva terrestre.
<i>Raster</i>	Caracteriza-se pela representação dos dados sob a forma de células

<i>Topology</i>	Representação, no espaço, de objetos e o relacionamento entre si.
-----------------	---

2.2.1.1 *Geometry*

Através deste sistema de representação, é possível todos os elementos de um mapa através de 3 estruturas: Ponto, linha e polígono.

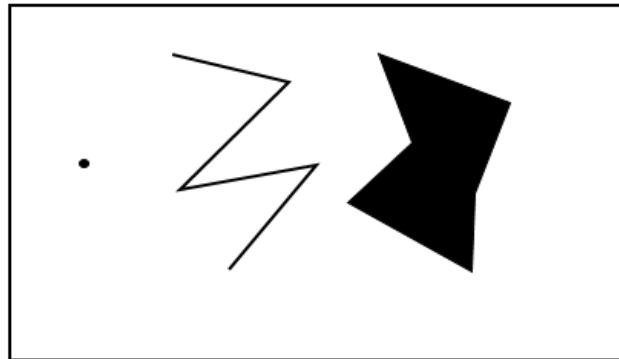


Figura 1: Ponto, linha e polígono

Por exemplo, uma cidade pode ser representada por um Polígono, um motorista por um Ponto e o caminho a percorrer por uma linha. Com esta representação é possível saber se um motorista se encontra numa determinada cidade ou se irá atravessar a mesma no seu percurso.

A *Geometry* trata o mundo como um plano cartesiano, sendo os cálculos realizados para a obtenção de distâncias, áreas, entre outros, obtidos através de geometria analítica básica.

2.2.1.2 *Geography*

Apesar de a *Geometry* ser uma representação funcional no que toca a pequenas regiões, esta pode tornar-se limitada quando a realização de cálculos requer que seja tida em conta a curvatura da Terra.

A representação e o tratamento dos dados através deste tipo é feita com o recurso a coordenadas esféricas. Assim, um ponto é aqui representado pelo ângulo de rotação a partir de um meridiano de referência (longitude) e o ângulo do equador (latitude).

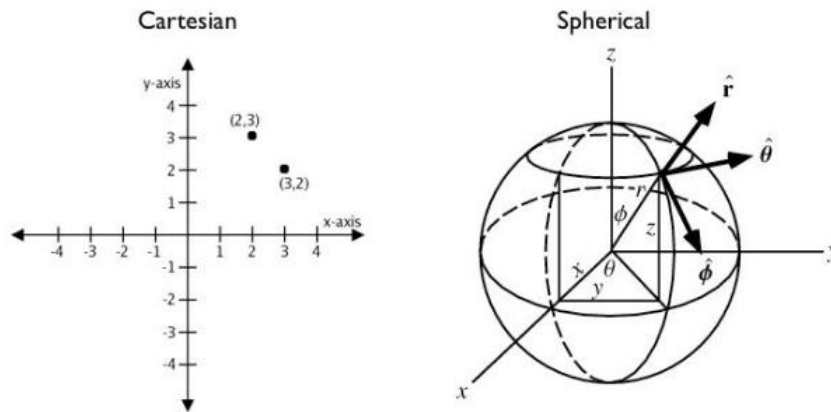


Figura 2: Representação cartesiana e esférica

Neste tipo de representação geográfica, os dados podem ser tratados como coordenadas cartesianas. Contudo, uma vez que a representação esférica mede a distância angular, as unidades são em graus.

2.2.1.3 Raster

Geometry e Geography são representações baseadas em vetores. Apesar de serem alternativas para a representação de coordenadas geográficas, o mesmo não se aplica quando a necessidade passa pela representação de mapas de temperaturas ou altitudes. É para estes casos que surge o tipo de representação *Raster*. Assim, o *Raster* representa um mosaico de *pixels* em que cada unidade armazena três cores diferentes (vermelho, verde e azul), designadas *bands*.

Um exemplo de utilização da representação *Raster* é na identificação da distribuição de vegetação global, como é apresentado na Figura 3.

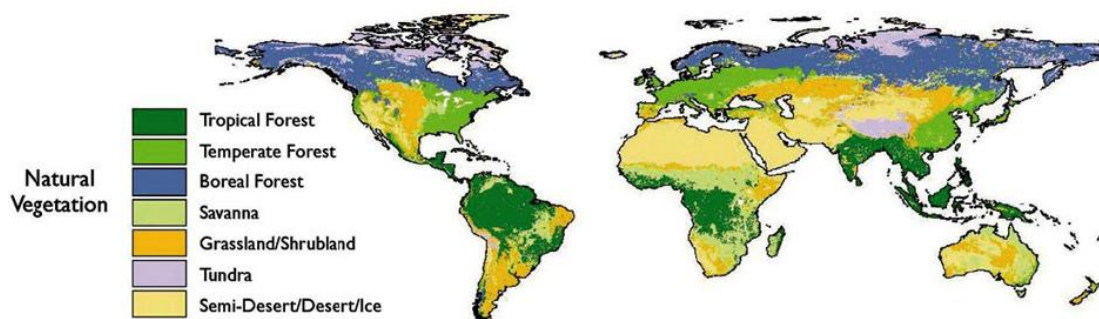


Figura 3: Distribuição global de vegetação

Este tipo de representação é também utilizado noutras aplicações:

- Mapas de temperatura e variação de altitude;
- Fotos de satélite (Esta representação pode ser constituída por 4 *bands* em cada pixel).

2.2.1.4 Topology

Ao contrário dos tipos de representação até aqui abordados (*Geometry*, *Geography* e *Raster*), o *Topology* não se caracteriza por disponibilizar funcionalidades relacionadas com formas e localizações geográficas. O *Topology* é direcionado para a forma como as formas estão relacionadas entre si.

Por exemplo, no projeto de arquitetura de uma habitação, em que é utilizada uma base de dados para a representação espacial das divisões da mesma, podem ser utilizados os seguintes tipos de dados:

- Linhas para a representação da divisão de uma sala entre sala de estar e sala de jantar;
- Polígonos para a representação das divisões da habitação (sala, cozinha, quartos, entre outras).

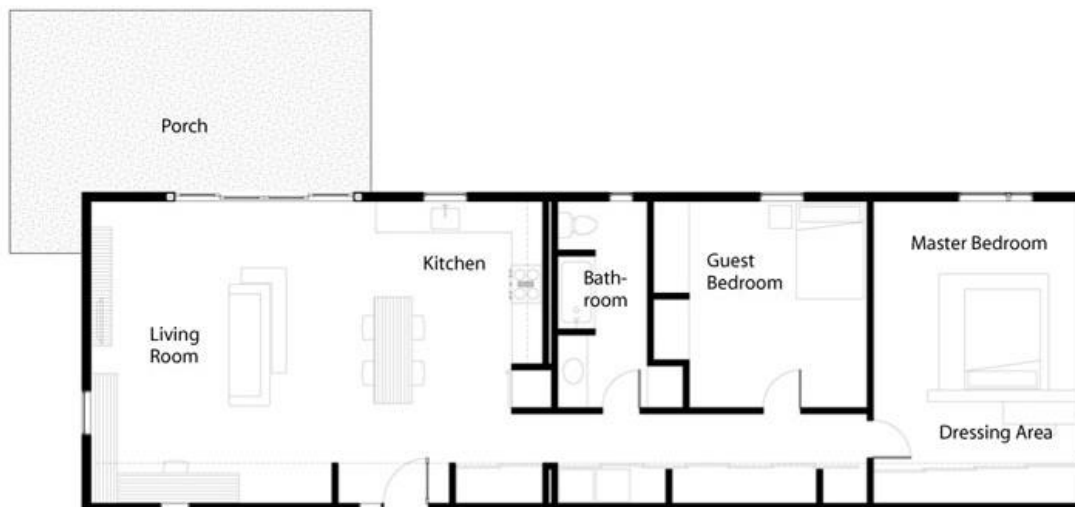


Figura 4: Planta de habitação

No plano representado pela Figura 4, para alterar o tamanho do quarto principal e reduzir o tamanho do quarto de hóspedes (através da movimentação da parede que os separa) teriam de ser realizadas duas operações no caso da utilização da *Geometry* e apenas uma na utilização da *geography*.

2.2.2 Tipos de dados

O PostgreSQL, através da extensão PostGIS, disponibiliza um conjunto de tipos de dados que auxiliam na representação geométrica da informação espacial. Nesta secção do documento são apresentados três dos vários tipos disponibilizados pela extensão.

2.2.2.1 Ponto

Um ponto, na extensão PostGIS, pode assumir vários subtipos. A Tabela 8 apresenta todos os subtipos de pontos existentes bem como uma descrição de cada um deles.

Tabela 8: Subtipos de Ponto

Subtipo	Descrição
POINT	Ponto bidimensional especificado pelas coordenadas X e Y.
POINTZ	Ponto tridimensional especificado pelas coordenadas X, Y e Z.
POINTM	Ponto bidimensional com uma medida especificado pelas coordenadas X e Y e pela medida M.
POINTZM	Ponto tridimensional especificado pelas coordenadas X, Y, Z e pela medida M.

2.2.2.2 Linha

Uma linha é caracterizada pela ligação de dois ou mais pontos. A ligação individual entre os pontos designa-se segmento. Os segmentos não representam tipos de dados no PostGIS contudo, é possível obter um segmento de uma linha.

De forma semelhante aos Pontos, as linhas têm quatro subtipos apresentados na Tabela 9.

Tabela 9: Subtipos de Linha

Subtipo	Descrição
LINestring	Linha bidimensional especificada por dois ou mais POINTs distintos.
LINestringZ	Linha tridimensional especificada por dois ou mais POINTZs distintos.
LINestringM	Linha bidimensional especificada por dois ou mais POINTMs distintos.
LINestringZM	Linha tridimensional especificada por dois ou mais POINTZMs distintos.

Associado a este tipo de dados está ainda o conceito de linha aberta ou fechada. Ambas são constituídas por múltiplos pontos contudo, numa linha fechada, o primeiro e último pontos são o mesmo.

2.2.2.3 Polígono

Um polígono é definido por uma lista de pontos correspondentes aos seus vértices, em que o último ponto é o primeiro, formando assim uma linha fechada.

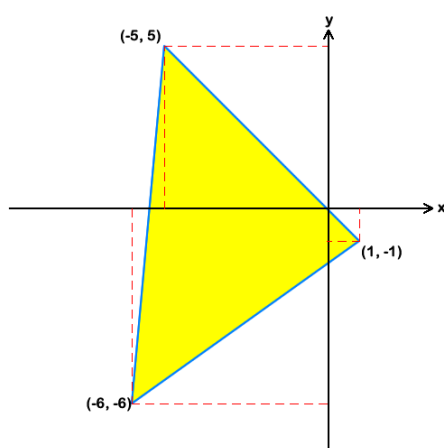


Figura 5: Representação de um polígono e dos pontos utilizados para sua definição

2.2.3 Funções de acesso aos dados

A extensão PostGIS disponibiliza um vasto catálogo de funções, organizadas em várias categorias, que permitem a gestão do modelo de dados, a inicialização de tipos de dados espaciais bem como outras funções auxiliares. Na Tabela 10 é apresentado um conjunto de funções utilizadas no trabalho apresentado neste documento.

Tabela 10: Funções do PostGIS

Função	Descrição
AddGeometryColumn	Adiciona uma coluna para o registo de dados do tipo Geometry a uma tabela existente.
ST_GeogFromText	Devolve um objeto geográfico definido sob a forma de texto.
ST_GeometryFromText	Devolve um objeto geométrico definido sob a forma de texto.
ST_LineFromText	Devolve uma linha definida a partir de texto.
ST_MakeEnvelope	Cria um polígono retangular a partir dos pontos passados nos argumentos.
ST_AsText	Devolve um objeto geométrico/geográfico representado no formato de texto.
ST_Distance	Quando invocada com objetos geométricos, devolve a distância entre eles. Para objetos geográficos calcula e devolve a distância geodésica mínima entre eles.
ST_Within	Devolve verdadeiro no caso de o objeto geométrico A se encontrar dentro do objeto geométrico B.

2.2.4 Operadores

Um conjunto de novos operadores é disponibilizado pelo PostGIS. A Tabela 11 abaixo apresenta uma lista com alguns dos operadores disponibilizados pelo PostGIS.

Tabela 11: Operadores

Operador	Descrição
&&	Verdadeiro no caso do objeto A interseção o objeto B. É aplicável a dados bidimensionais.
&&&	É uma extensão ao operador &&, contudo permite fazer a comparação de objetos n-dimensionais.
&<	Verdadeiro no caso do objeto A se sobrepor ou estar à esquerda do objeto B.
&<	Verdadeiro no caso do objeto A se sobrepor ou estar abaixo do objeto B.
&>	Verdadeiro no caso do objeto A se sobrepor ou estar à direita do objeto B.
<<	Verdadeiro no caso do objeto A estar estritamente à esquerda do objeto B.
<<	Verdadeiro no caso do objeto A estar abaixo do objeto B.
=	Verdadeiro no caso do objeto A ser o mesmo do objeto B.
>>	Verdadeiro caso o objeto A estar à direita do objeto B.
@	Verdadeiro se o objeto B incluir o objeto A.
&>	Verdadeiro se o objeto A se sobrepor ou estiver acima do objeto B.
>>	Verdadeiro se o objeto A estiver acima do objeto B.
~	Verdadeiro se A contém B.
~=	Verdadeiro se A é igual a B.
<->	Devolve a distância bidimensional entre A e B.

2.3 Elasticsearch

O Elasticsearch é uma alternativa não relacional ao armazenamento de dados. Dentro dos tipos de Bases de Dados não relacionais abordados na secção 2.1.2, esta caracteriza-se como uma *document database*.

Ao contrário de uma base de dados relacional, o Elasticsearch indexa, por omissão, todos os campos pelo que, a única cuidado a ter na criação de *schemas* passa pela definição correta dos tipos de dados de modo a que posteriormente se possa tirar proveito na realização das pesquisas.

2.3.1 Tipos de dados

Ao contrário do PostgreSQL, o Elasticsearch não requer a instalação de extensões de forma a disponibilizar tipos de dados espaciais. Esta secção apresenta os tipos de dados disponibilizados pela instalação *standard* do Elasticsearch.

2.3.1.1 Geo-point

Este tipo de dados serve o registo de pares latitude-longitude que podem ser utilizados em operações como:

- Encontrar todos os pontos dentro de uma área definida, os pontos mais próximos de um segundo ponto, os pontos dentro de uma determinada distância, entre outros;
- Agregar os resultados pela distância a um determinado ponto;
- Utilizar a distância a um determinado ponto na pontuação atribuída aos resultados da pesquisa;
- Ordenar os pontos pela distância a um determinado ponto.

2.3.1.2 Geo-shape

O tipo de dados *Geo-shape* facilita a indexação e a pesquisa de estruturas como retângulos e polígonos. Não oferece 100% de precisão e, dependendo da sua configuração, pode retornar alguns falsos positivos/falsos negativos para as pesquisas realizadas. Para atenuar este comportamento, o Elasticsearch permite a definição de um parâmetro para ajustar as expectativas dos resultados.

2.3.2 Funções

Nesta seção são apresentadas algumas funções de pesquisa disponibilizadas pelo Elasticsearch.

Tabela 12: Funções disponibilizadas pelo Elasticsearch

Função	Descrição
geo_bounding_box	Permite a filtragem dos resultados com o recurso à utilização de um perímetro.
geo_distance	Filtra os resultados da pesquisa com base num ponto e na distância especificada a esse ponto.
geo_polygon	Inclui, nos resultados da pesquisa, apenas aqueles que se encontram dentro do polígono especificado.
geo_distance	Permite a agregação dos resultados da pesquisa em séries de distância.

3 Desempenho

O presente capítulo faz uma apresentação da primeira parte do projeto realizado, isto é, da avaliação do desempenho das Bases de Dados espaciais. Neste tópico é caracterizada a máquina onde foram realizados os testes bem como as tecnologias avaliadas (secção 3.1), os modelos de dados criados (secção 3.2), descrita a forma de obtenção de dados que serviu de suporte à realização dos testes (secção 3.3), os resultados obtidos nas pesquisas realizadas bem como o consumo de memória pelas Bases de Dados (secções 3.4 e 3.5) e, por fim, as conclusões retiradas dos testes realizados (secção 3.6).

3.1 Ambiente

A avaliação do desempenho das bases de dados foi realizada numa máquina com um CPU Intel i7 2.9GHz com 16GB de memória e o sistema operativo Ubuntu 14.04.5 LTS 64-bit. Os testes foram realizados a uma Bases de Dados PostgreSQL 9.4.9 com a extensão PostGIS 2.2.2 e à base de dados Elasticsearch na versão 2.4.0. A inclusão de, apenas, estas duas Bases de Dados nos testes de desempenho baseia-se no facto de serem as tecnologias atualmente em uso na Jumia Services.

3.2 Modelo de dados

O primeiro passo para a avaliação do desempenho das Bases de Dados passou pelo desenho do modo de representação de informação. O modelo de dados utilizado neste caso de estudo ajusta-se às necessidades da Jumia Services, no entanto, foi desenhado de forma genérica com o objetivo de escalar para eventuais necessidades. Assim, o modelo de dados assenta nas seguintes entidades:

- **Driver** – Representa um motorista. Esta entidade contém as propriedades que identificam inequivocamente o motorista.

- **Entity** – Sendo um *software* de logística, que pretende monitorizar a transação de bens entre entidades, este objeto representa cada um dos bens transacionados. De forma semelhante aos *Drivers*, possui as propriedades que representam inequivocamente os bens.
- **Event** – Presente representar os eventos realizados pelos motoristas (entrega de um bem, pausa para almoço, abastecimento da viatura, etc.).
- **Location** – Entidade principal do modelo. Representa as localizações registadas ao longo do tempo pelos motoristas, bem como uma lista de meta dados recolhidos.

A Figura 6 apresenta as entidades criadas para a representação da informação bem como o relacionamento entre si.

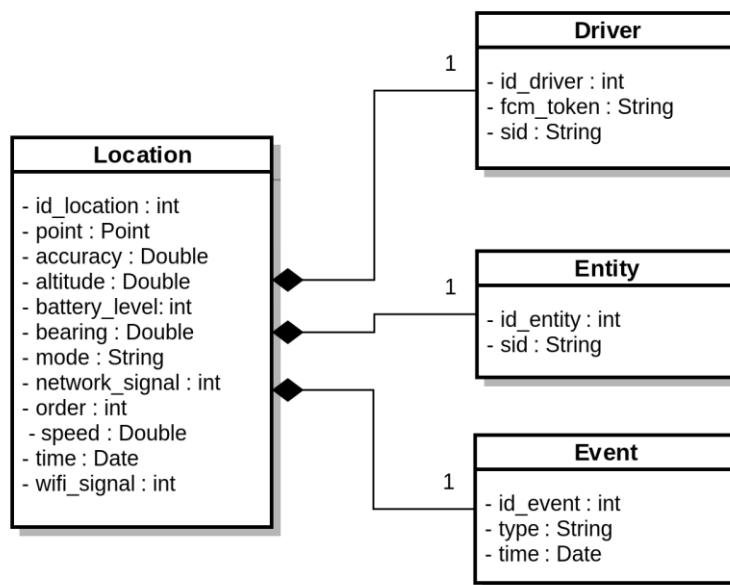


Figura 6: Modelo de dados

A representação de informação numa Base de Dados relacional é feita através de tabelas pelo que as tabelas a serem criadas podem corresponder às respetivas entidades. No entanto, a representação de dados numa base de dados NoSQL, em concreto no Elasticsearch, é feita através de documentos, não havendo relações entre os mesmos. Para ilustrar as representações nas diferentes bases de dados, são abaixo apresentadas imagens (Figura 7 e Figura 8) que representam a estrutura da informação no PostgreSQL e Elasticsearch respetivamente.

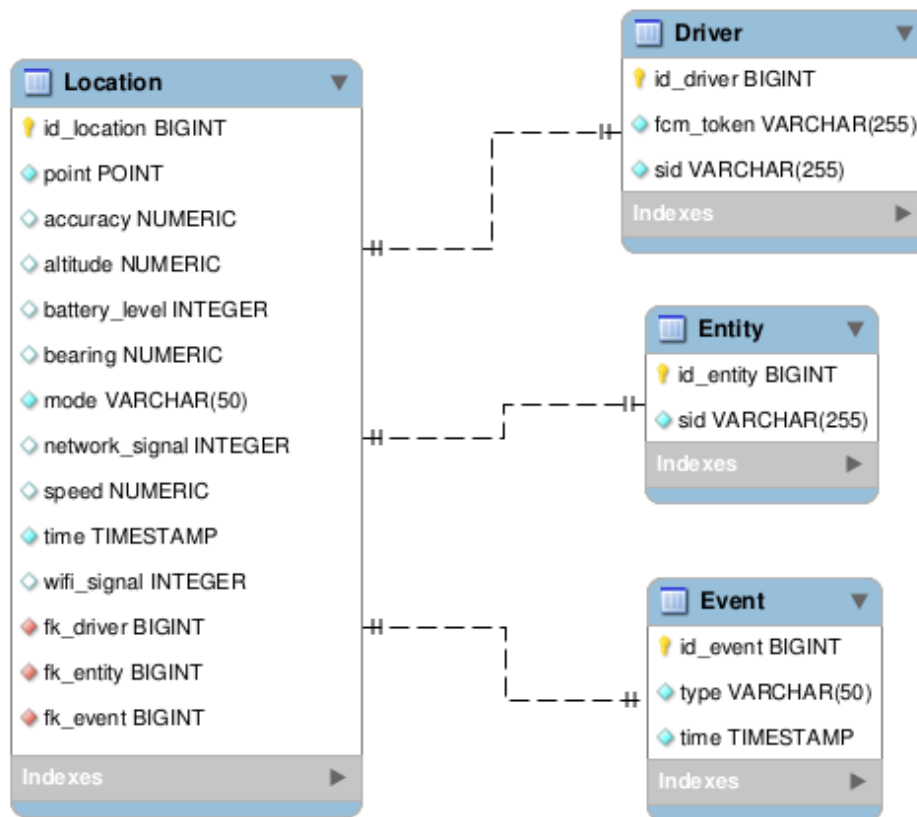


Figura 7: Diagrama de Entidade e Relacionamento

```

1
2
3 {
4   "tnt_v5_locations_eg": {
5     "mappings": {
6       "live": {
7         "properties": {
8           "accuracy",
9           "altitude",
10          "batteryLevel",
11          "bearing",
12          "driver": {
13            "properties": {
14              "fcmToken",
15              "id",
16              "sid"
17            }
18          },
19          "entity": {
20            "properties": {
21              "id",
22              "sid"
23            }
24          },
25          "event": {
26            "properties": {
27              "id",
28              "time",
29              "type"
30            }
31          },
32          "id",
33          "mode",
34          "networkSignal",
35          "point",
36          "speed",
37          "time",
38          "timeStr",
39          "wifiSignal"
40        }
41      }
42    }
43  }
44 }
45

```

Figura 8: Mapeamentos ElasticSearch

Depois de realizados alguns testes de desempenho, constatou-se que a base de dados PostgreSQL, pela sua natureza relacional, apresentou fracos resultados (quando comparada com o ElasticSearch) pela necessidade da relação entre tabelas para a obtenção da localização atual do motorista. Assim, optou-se por executar os mesmos testes em mais uma base de dados PostgreSQL. A estrutura desta base de dados é semelhante à apresentada na Figura 7, com a diferença de, na tabela de motoristas – Driver – ter um campo adicional com a localização atual do motorista. Assim, sempre que é adicionado um novo registo na tabela de localizações é

atualizada a localização do respetivo motorista. Nos resultados da realização dos testes, a identificação dos ambientes testados é:

- ElasticSearch para os testes realizados na base de dados ElasticSearch;
- PostgreSQL A para a base de dados PostgreSQL com o modelo de dados inicial;
- PostgreSQL B para a base de dados PostgreSQL com o modelo alterado.

Assim, os testes foram realizados em 3 bases de dados distintas:

- Não relacional: ElasticSearch;
- Relacional: PostgreSQL com os dois modelos distintos.

3.3 Universo de dados

Atendendo ao propósito da realização dos testes de desempenho das bases de dados espaciais no contexto do registo das localizações dos motoristas da Jumia Services, foi criado um universo de dados que permita a realização de testes a uma escala mais aproximada da realidade possível. Assim, a criação de um universo de dados para suportar as pesquisas foi dividido nos seguintes passos:

- Identificação das N cidades mais com mais população em Portugal;
- Simulação de rotas com as cidades identificadas no ponto anterior;
- Utilização das rotas para simular a entrega de encomendas por parte dos motoristas e inserção desses dados nas Bases de Dados. As simulações incluem a existência de eventos durante a viagem e a alteração de modos de viagem.

3.3.1 Identificação das cidades

O primeiro passo passou pela identificação das 125 cidades (apresentadas na Tabela 13) com mais população de Norte a Sul de Portugal. (“PORDATA - População residente segundo os Censos: total e por grandes grupos etários - Municípios,” n.d.)

Tabela 13: Cidades mais populadas de Portugal

Lisboa	Santarém	Ribeira Grande
Sintra	Figueira da Foz	Espinho
Vila Nova de Gaia	Felgueiras	Lagos
Porto	Alcobaça	Santiago do Cacém
Cascais	Évora	Fundão
Loures	Paços de Ferreira	Anadia
Braga	Amarante	Benavente
Matosinhos	Castelo Branco	Tondela
Amadora	Portimão	Peniche
Almada	Ovar	Estarreja

Oeiras	Pombal	Lamego
Gondomar	Marco de Canaveses	Montemor-o-Velho
Seixal	Vila Real	Tavira
Guimarães	Covilhã	Odemira
Odivelas	Caldas da Rainha	Lourinhã
Coimbra	Montijo	Albergaria-a-Velha
Santa Maria da Feira	Fafe	Portalegre
Vila Franca de Xira	Sesimbra	Seia
Maia	Vila Verde	Cartaxo
Vila Nova de Famalicão	Águeda	Porto de Mós
Leiria	Lousada	Mirandela
Setúbal	Ourém	Vizela
Barcelos	Olhão	Almeirim
Funchal	Ponte de Lima	Elvas
Viseu	Alenquer	Oliveira do Bairro
Valongo	Santa Cruz	Lagoa
Viana do Castelo	Guarda	Vale de Cambra
Paredes	Chaves	Vagos
Vila do Conde	Albufeira	Arcos de Valdevez
Torres Vedras	Tomar	Arouca
Barreiro	Abrantes	Salvaterra de Magos
Aveiro	Trofa	Póvoa de Lanhoso
Mafra	Marinha Grande	Machico
Penafiel	Ílhavo	Azambuja
Santo Tirso	Silves	São João da Madeira
Loulé	Torres Novas	Rio Maior
Ponta Delgada	Cantanhede	Vila da Praia da Vitória
Oliveira de Azeméis	Beja	Oliveira do Hospital
Moita	Câmara de Lobos	Baião
Faro	Angra do Heroísmo	Mealhada
Póvoa de Varzim	Bragança	Cinfães
Palmela	Esposende	

As cidades acima apresentadas serviram de ponto de partida para o desenho de rotas, abordado na secção 3.3.2.

3.3.2 Desenho de rotas

O segundo ponto passou pela criação das rotas, que serviram de simulação das viagens realizadas pelos motoristas para a entrega de encomendas. Para isso, foram realizadas todas as combinações possíveis entre todas as cidades listadas na Tabela 13. Os pares (cidade origem – cidade destino) resultantes das combinações deram origem a 15 500 rotas.

Para a obtenção de rotas, com coordenadas geográficas, foi utilizada a API *Directions* da Google. Esta API calcula as direções entre uma origem e um destino, permitindo ainda especificar o modo de viagem (a pé, de carro, etc.), os pontos de passagem, entre outros. Para este projeto, na simulação das rotas, em cada um dos pedidos foram especificados apenas a cidade de origem, destino e modo de viagem, como se pode verificar na Figura 9.

```

1 {
2   "origin": "Porto, PT",
3   "destination": "Lisboa, PT",
4   "travelMode": "DRIVING"
5 }
6

```

Figura 9: Corpo do pedido para a rota entre duas cidades

Os resultados de cada um dos pedidos realizados à API *Directions* foi armazenado numa base de dados auxiliar. Assim, num passo seguinte, todos os dados estiveram disponíveis para uma simulação, mais próxima da realidade, das rotas realizadas pelos motoristas da empresa.

No que toca à informação dos motoristas e das encomendas (tabela *Entity* da base de dados) foram gerados valores aleatórios para os identificar. No caso dos motoristas, foram gerados 100 identificadores. No caso das encomendas, foi gerado um registo por cada viagem.

3.3.3 Simulação das viagens

Depois de obtidas todas as rotas da API *Directions* da Google constatou-se que, em média, cada rota é constituída por 240 pontos geográficos. Assim, no sentido de fazer corresponder cada rota à entrega de uma encomenda, optou-se por dividir cada uma em 3 fases. A Figura 10, seguida de uma explicação, apresenta o exemplo de uma simulação de viagem.



Figura 10: Simulação de viagem

- Numa fase inicial, representado a verde na imagem (entre o ponto 0 e o ponto 42), a viagem é realizada em *slow mode*, com os pontos a serem registados com uma diferença de 5 minutos do anterior;
- No ponto 42 o motorista dá início à entrega de uma encomenda, com o registo de um evento. É também a partir de este ponto que a viagem passa a ser feita em *emergency mode*, sendo os pontos registados com 5 segundos de diferença entre si;
- No ponto 190 o motorista termina a entrega da encomenda com o registo de um novo evento. A partir deste ponto volta a fazer a viagem em *slow mode*, de forma semelhante à primeira fase da viagem.

A escolha dos pontos de início e fim de entrega de uma encomenda é feita de forma aleatória. Para isso, é escolhido aleatoriamente um valor entre 25 e 55 e outro entre 180 e 210 para o ponto inicial e final respetivamente.

3.4 Análise de desempenho

A avaliação do desempenho das duas bases de dados foi realizada tendo por base o tempo necessário para a indexação de informação e também o tempo despendido na realização de pesquisas. Nas subsecções 3.4.1 e 3.4.2 são apresentados os resultados dos testes de escrita e leitura respetivamente.

3.4.1 Escrita

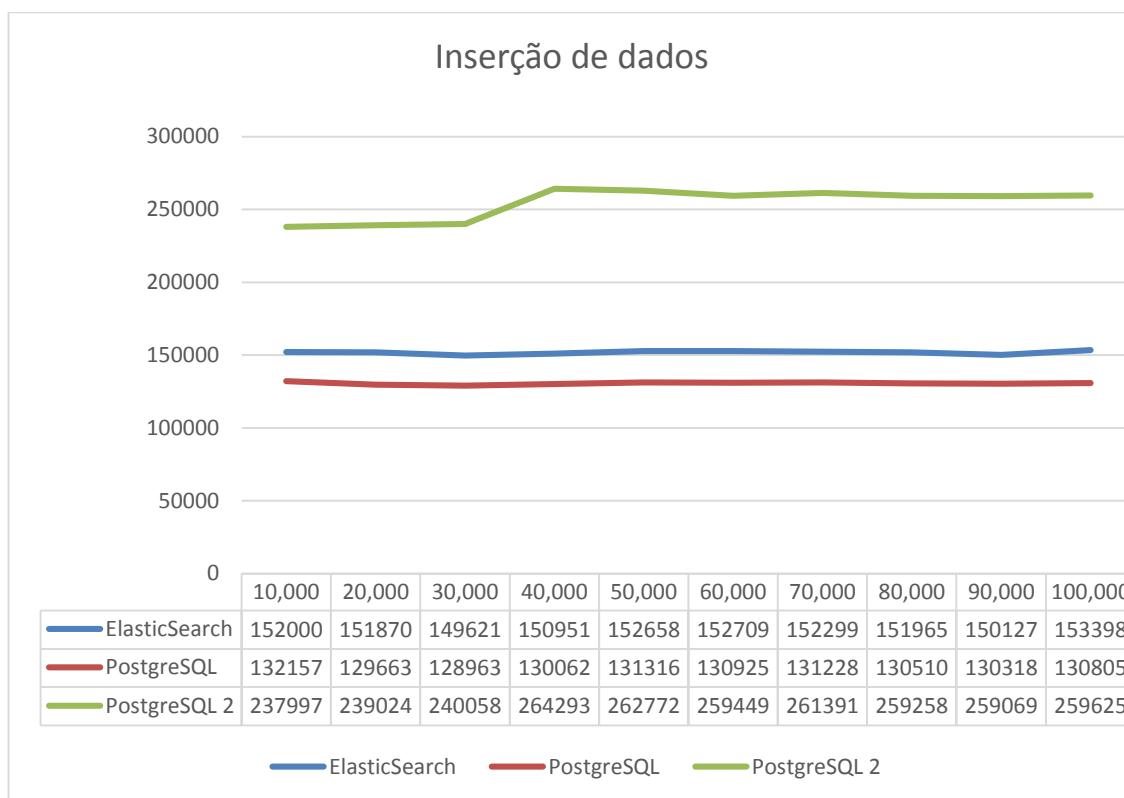


Gráfico 1: Registo de dados

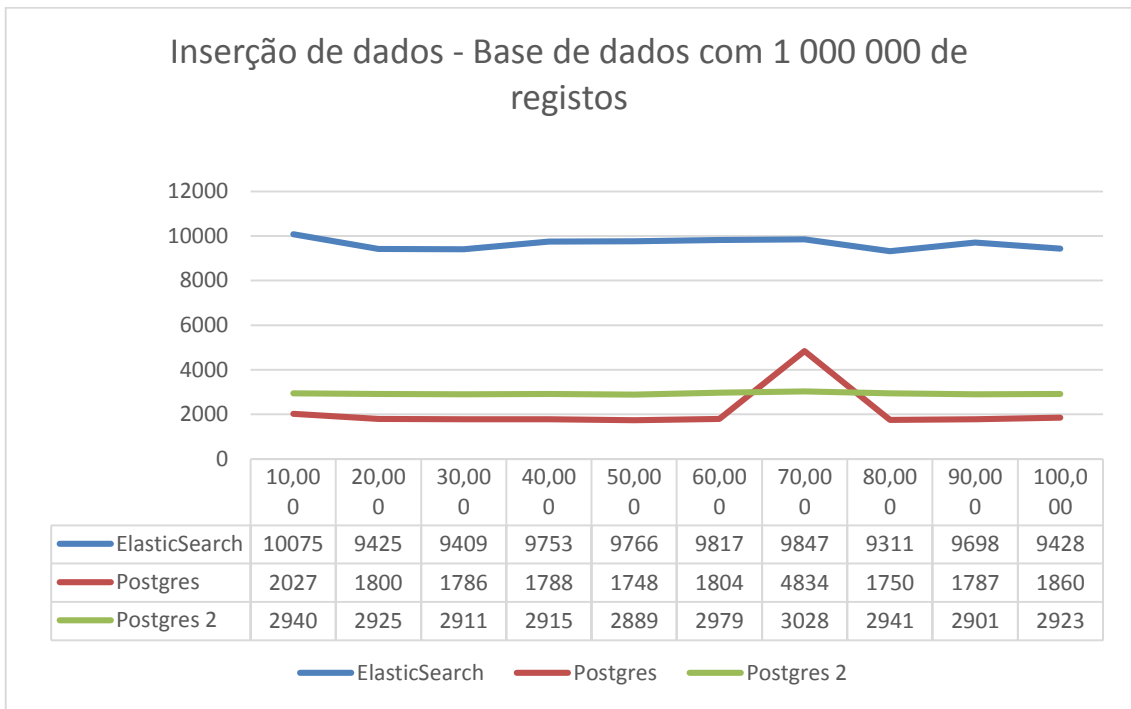


Gráfico 2: Teste de escrita – Base de dados com 1 000 000 de registos

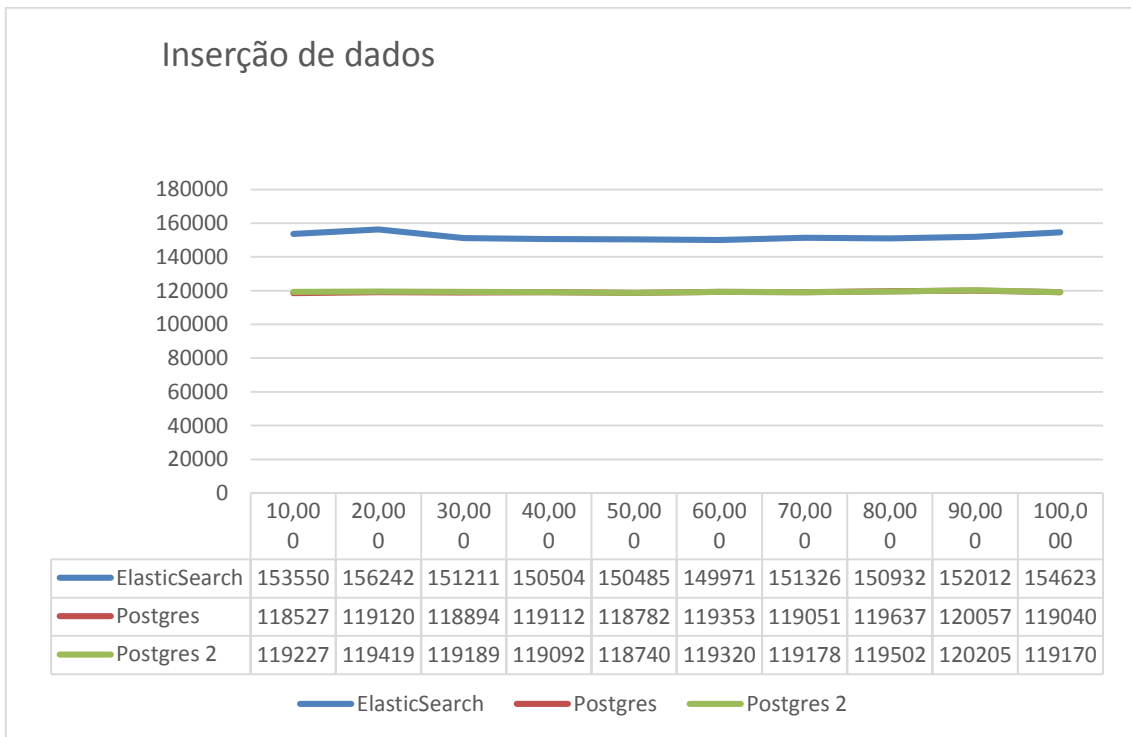


Gráfico 3: Inserção em bulk

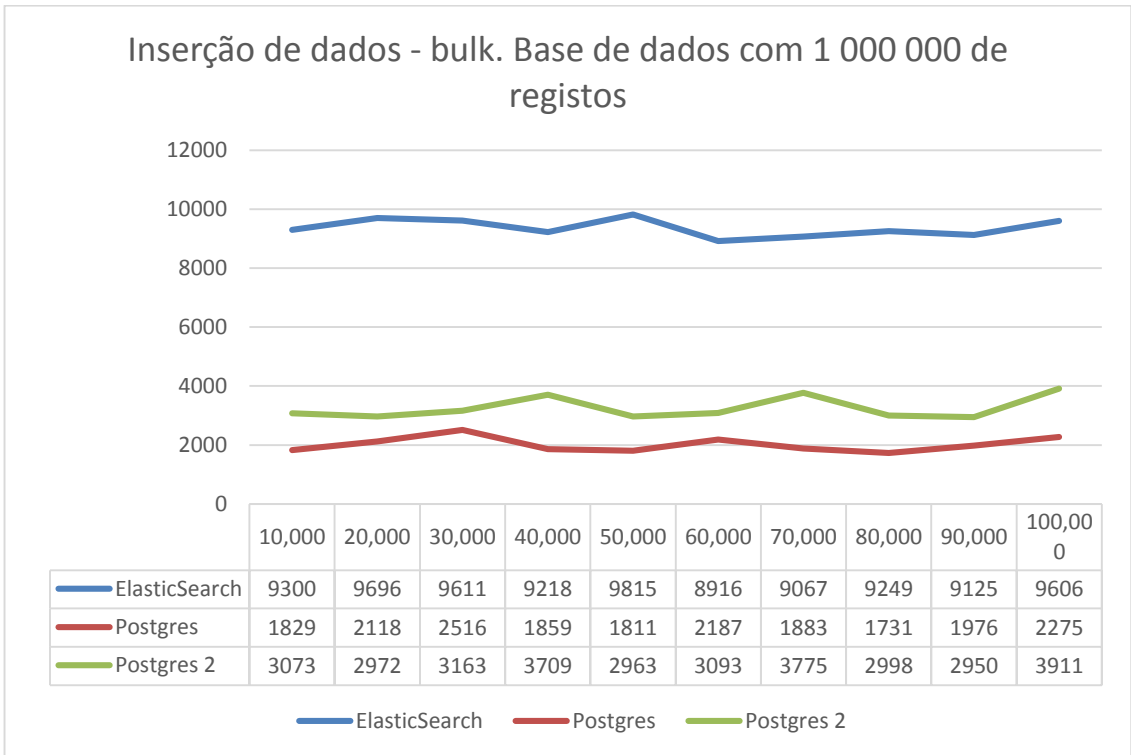


Gráfico 4: Inserção em bulk. Base de dados com 1 000 000 de registos

Como podemos avaliar pelos gráficos, as bases de dados apresentam um melhor desempenho quando inserem os dados em *bulk*. Quanto ao conteúdo da base de dados, através dos testes realizados, não foi encontrada uma diferença significativa na inserção de dados em bases de dados vazias ou com dados. No sentido de melhor compreender as diferenças, é apresentado um último gráfico que ilustra a média do tempo gasto por cada uma das bases de dados nas operações realizadas.

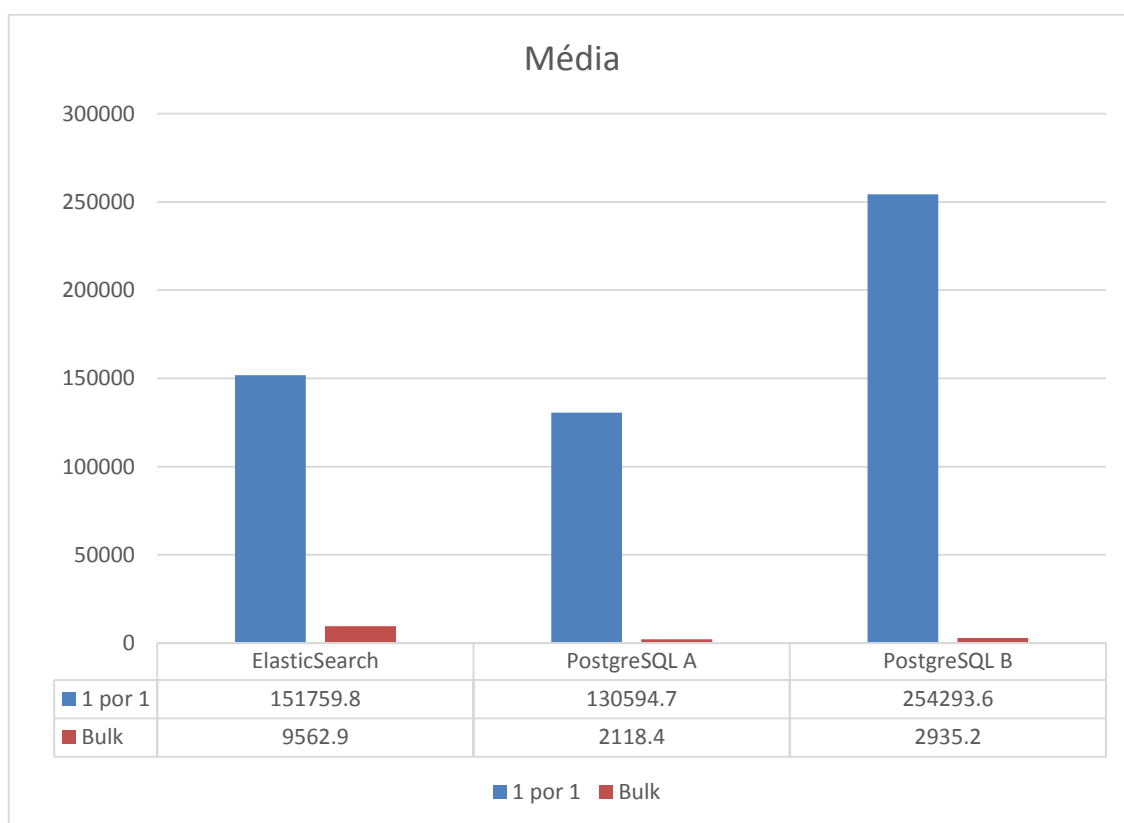


Gráfico 5: Média do tempo consumido na inserção de dados

Pela análise do gráfico notamos uma clara vantagem, em todas as bases de dados, no registo de dados em modo *bulk*. Quando comparadas a melhor solução do PostgreSQL e o ElasticSearch, notamos que o PostgreSQL surge com o melhor desempenho no registo de informação. Dentro dos diferentes modelos de dados do PostgreSQL, o modelo mais eficiente é o modelo A, uma vez que o B, em cada operação de inserção de um novo registo, necessita também de proceder à atualização de outro registo.

3.4.2 Leitura

No que toca a obtenção de informação nas bases de dados, o primeiro conjunto de testes pretende analisar o comportamento das bases de dados perante um conjunto de pesquisas básicas.

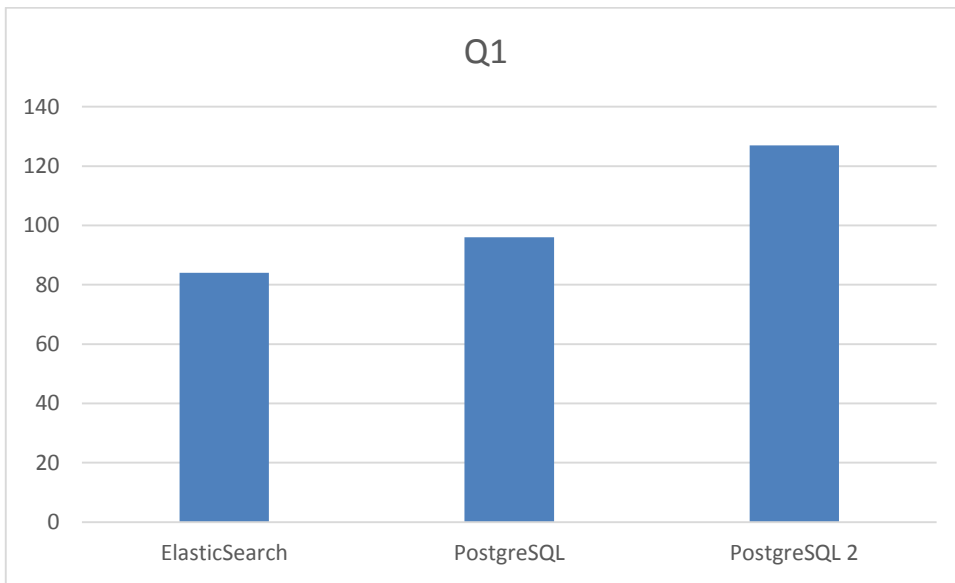


Gráfico 6: Total de registros na tabela de localizações

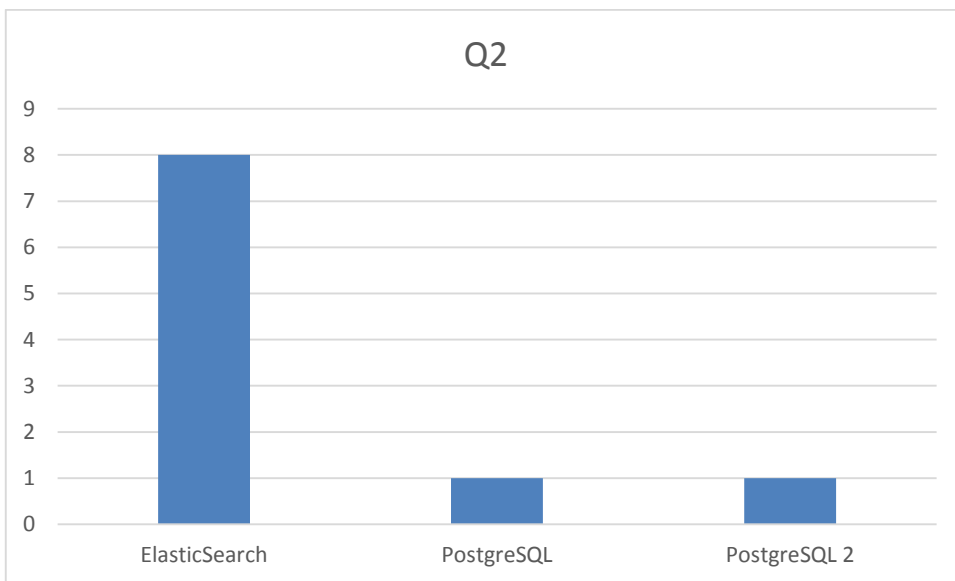


Gráfico 7: Obtenção de 200 localizações

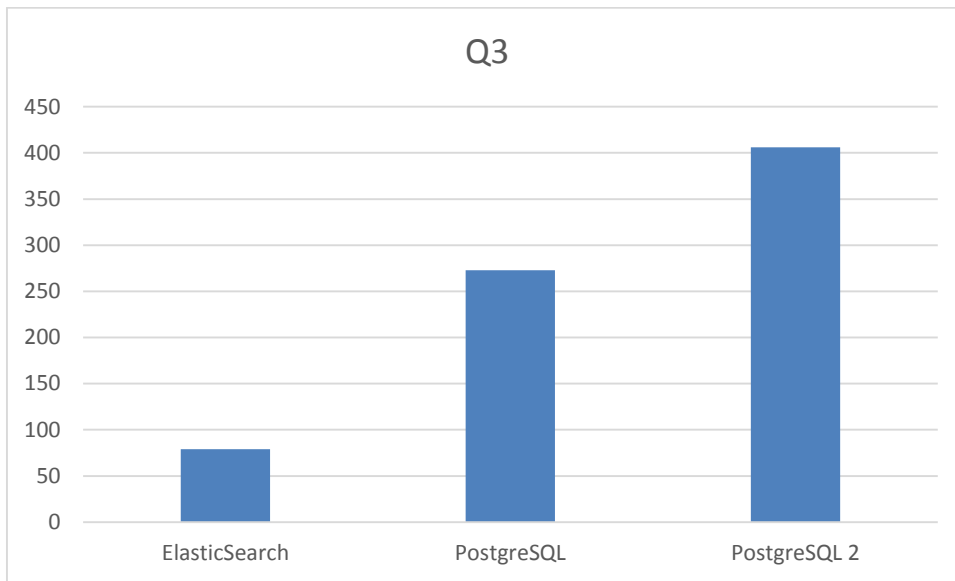


Gráfico 8: Número de localizações por motorista

Atendendo aos primeiros testes realizados, com a exceção da obtenção de 200 registos da base de dados, o ElasticSearch foi a solução que melhor desempenho apresentou.

Seguem-se o segundo conjunto de testes de leitura, onde foram realizados testes mais aproximados das necessidades da Jumia Services. Cada um destes testes foi realizado 5 vezes.

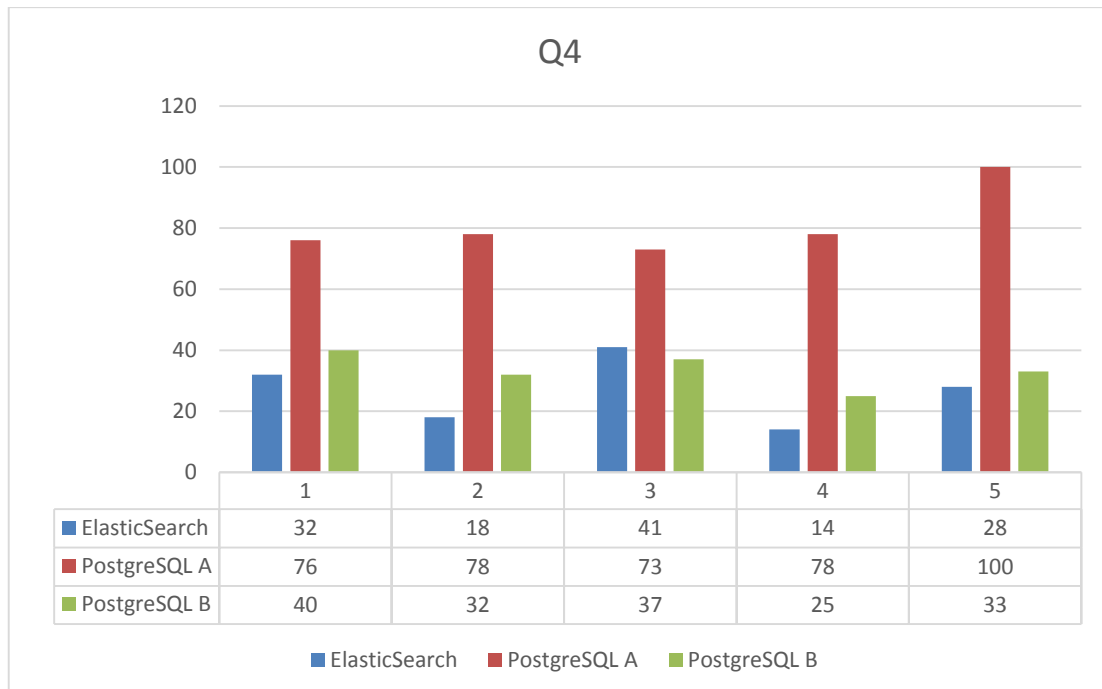


Gráfico 9: Leitura da última localização de cada um dos motoristas

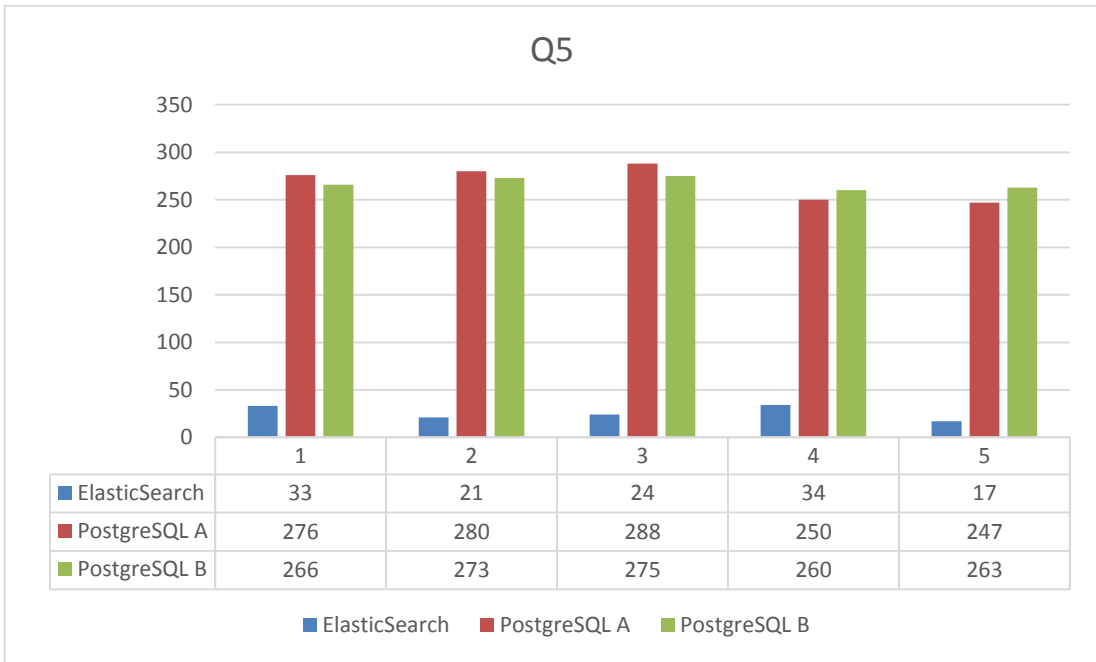


Gráfico 10: Localização atual dos motoristas e últimos 10 pontos

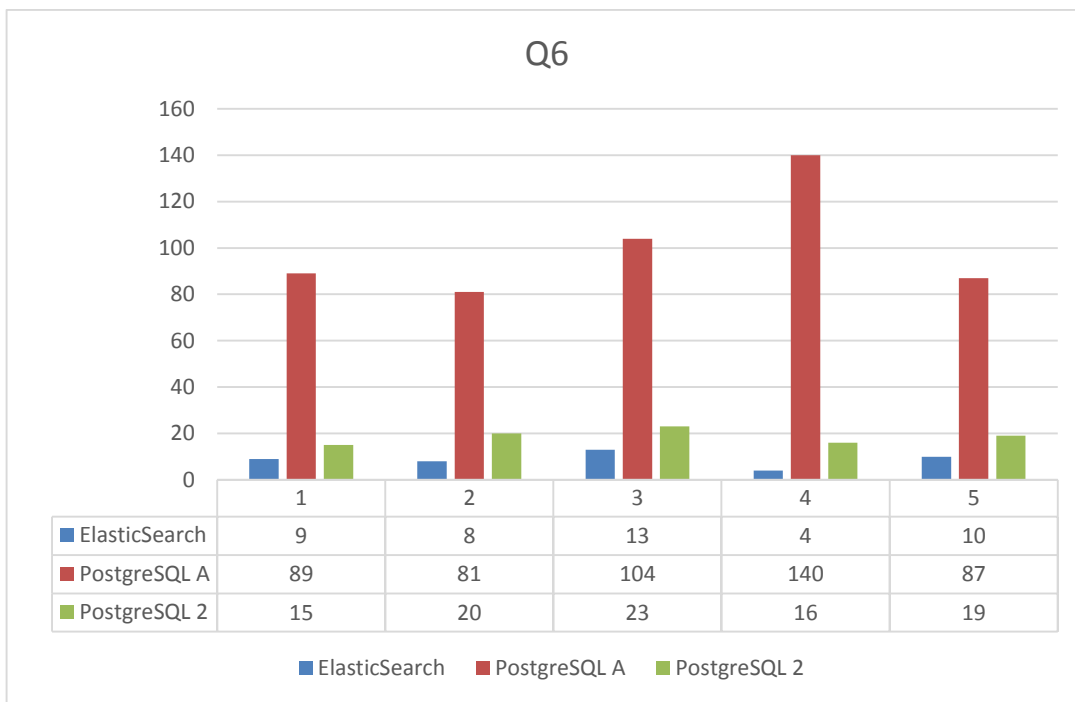


Gráfico 11: Localização de motoristas dentro de um retângulo

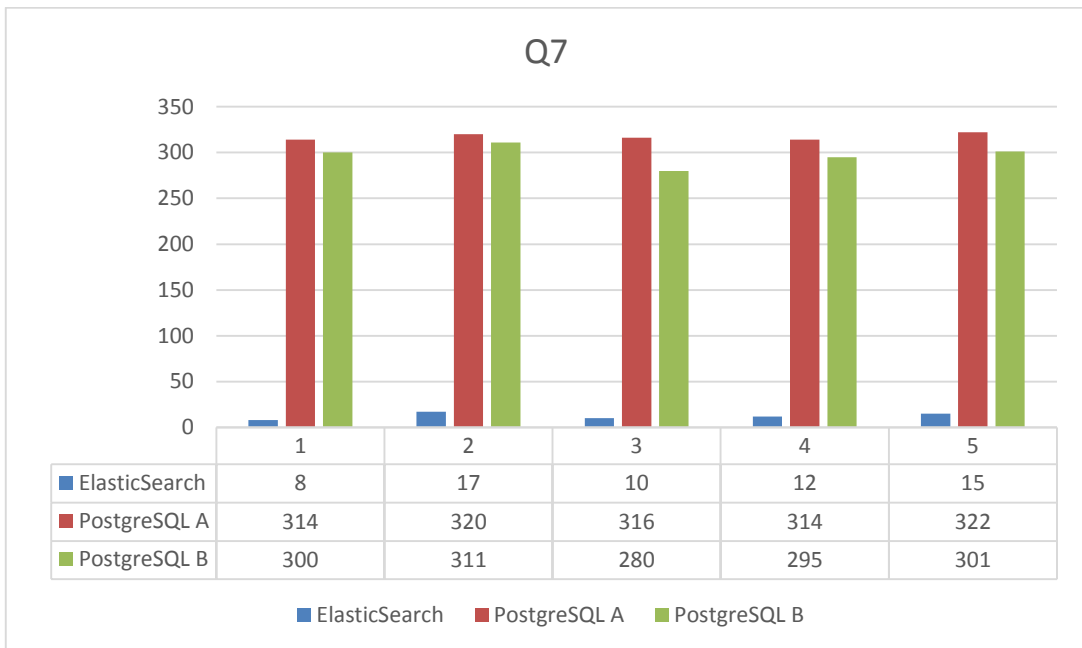


Gráfico 12: Localização atual dos motoristas dentro de um quadrado e últimos 10 pontos

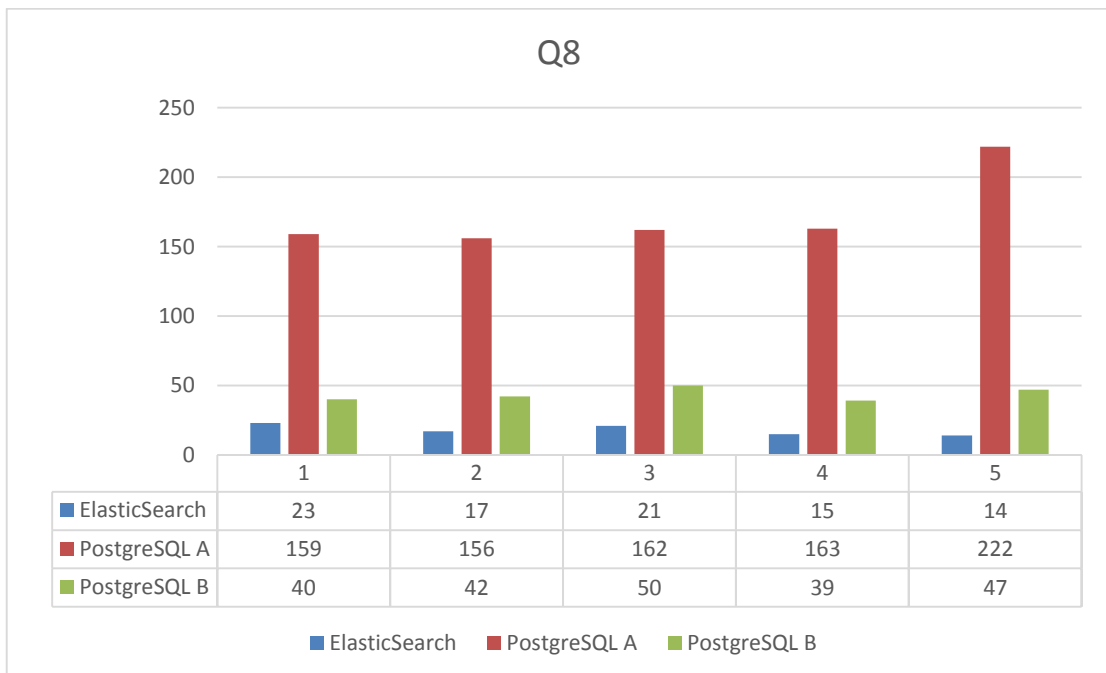


Gráfico 13: Localização de motoristas dentro de uma circunferência

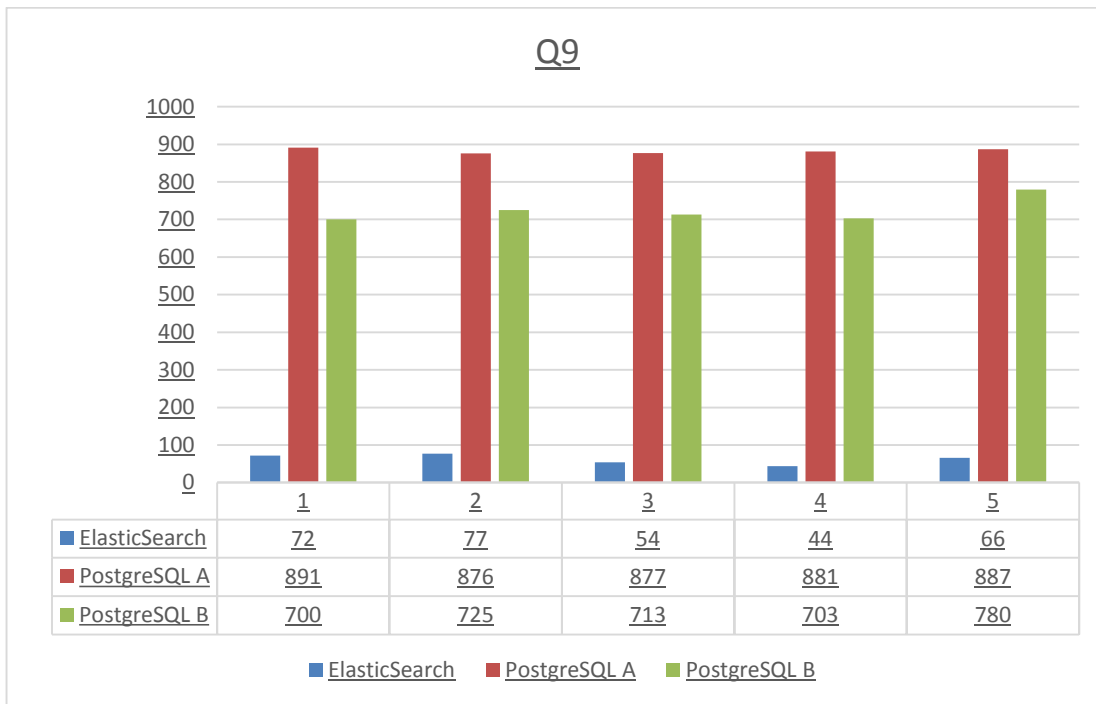


Gráfico 14: Localização de motoristas dentro de uma circunferência e últimos 10 pontos

De um modo geral, o ElasticSearch apresentou um melhor desempenho em todas as pesquisas, referentes ao segundo conjunto, realizadas às bases de dados.

3.5 Análise do espaço utilizado pelas bases de dados

Sendo as bases de naturezas distintas, com diferentes formas de indexação de informação, foi também realizado um levantamento do espaço utilizado pelas diferentes bases de dados analisadas. No Gráfico 15 é apresentado um gráfico com o espaço ocupado em disco por cada uma das bases de dados. Os valores aqui apresentados estão em MB.

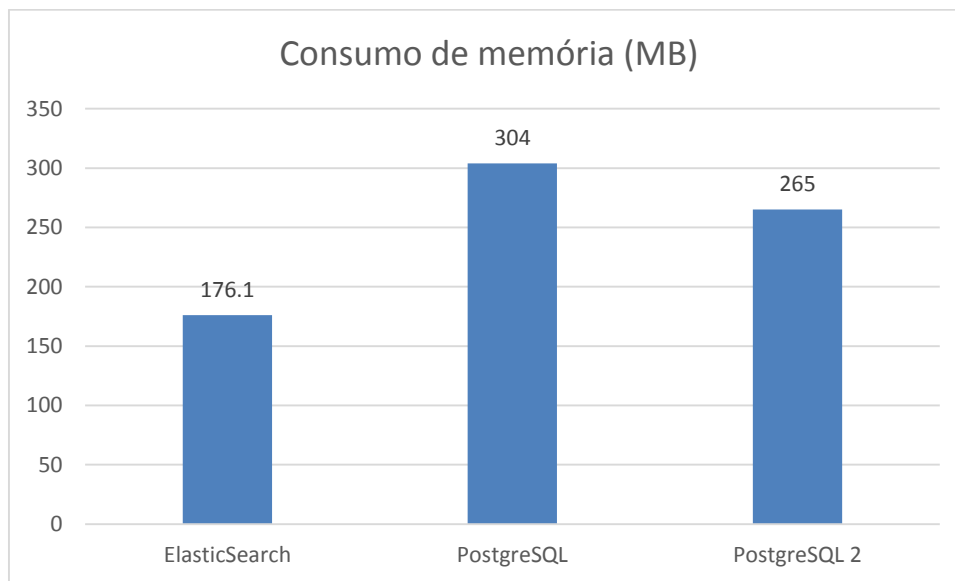


Gráfico 15: Consumo de memória pelas bases de dados

3.6 Conclusões

Neste estudo foram analisados os tempos necessários para realizar o mesmo tipo de pesquisas nas diferentes bases de dados, utilizando os casos de uso desenhados para a implementação de uma aplicação de monitorização, em tempo real, de motoristas. Tratando-se de uma aplicação de análise em tempo real, foram realizados testes de análise de desempenho na escrita e leitura de informação.

Quando analisados os resultados, percebemos que o PostgreSQL e o ElasticSearch apresentam resultados distintos em ambos os testes de escrita e de leitura. No que toca à escrita de informação nas bases de dados, é possível retirar duas conclusões principais:

- A escrita de dados, nas bases de dados em que foram realizados os testes, é mais eficiente quando é feita em modo *bulk*;
- A criação de novos registos é realizada de forma mais eficiente nas bases de dados PostgreSQL comparativamente com o ElasticSearch. De entre os modelos de dados do PostgreSQL estudados, o modelo A revela-se mais eficiente.

No que toca à leitura de informação da base de dados, existe uma solução que é claramente mais vantajosa quando comparada com as restantes. De um modo geral, nos testes realizados, o ElasticSearch demorou menos 30% do tempo despendido pelo PostgreSQL para a realização das pesquisas. Com os testes realizados é ainda possível concluir o seguinte:

- A obtenção dos últimos N registos de cada motorista revela-se uma pesquisa substancialmente mais demorada, para o PostgreSQL, quando comparada com a pesquisa da localização atual. No ElasticSearch ambas as pesquisas são realizadas em

tempos muito semelhantes. De entre os diferentes modelos do PostgreSQL, o modelo B é mais eficiente nas pesquisas não devolvem os últimos 10 pontos de cada motorista;

- A obtenção de todos os motoristas localizados dentro de um quadrado é feita de forma mais eficiente quando comparada com a pesquisa dos motoristas dentro de um círculo. Este facto verifica-se em ambas as bases de dados;
- A eficiência da pesquisa de pontos dentro de uma circunferência baixa com o aumento do raio. Este efeito verifica-se em ambas as bases de dados.

Pela análise dos resultados, podemos concluir que, em todas as pesquisas em que é necessário agrupar informação, como as localizações por motorista, o Elasticsearch demora substancialmente menos tempo para realizar as operações de leitura.

Ainda em termos de memória verificamos que o Elasticsearch utiliza menos espaço para o armazenamento da informação.

4 Rastreio de motoristas

O presente capítulo pretende apresentar, em detalhe, o desenvolvimento do protótipo. Desde a apresentação da arquitetura da aplicação (secção 4.1), passando pela análise e o *design* da solução (secções 4.2 e 4.3) e terminando no detalhe do desenvolvimento do protótipo (secção 0), todos os passos são apresentados.

4.1 Arquitetura

Apesar de recente, o HMT tem já implementado um vasto leque de requisitos funcionais e não funcionais. Desta forma, a aplicação está inserida numa arquitetura complexa que inclui diversos componentes.

Na Figura 11 é apresentado o diagrama de implementação da aplicação.

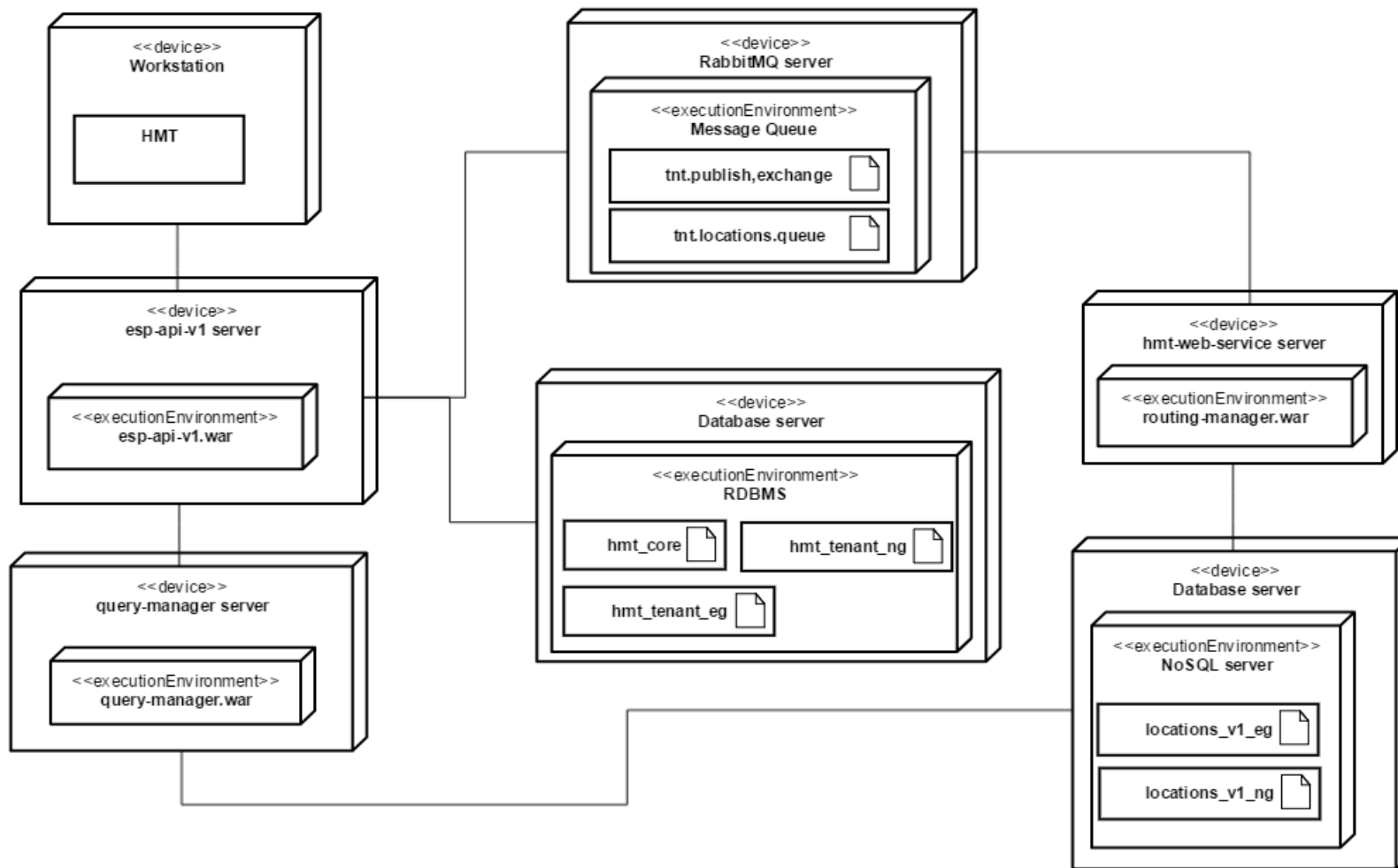


Figura 11: Diagrama de implementação

Como podemos verificar no diagrama apresentado na Figura 11, a infraestrutura que serve o HMT é composta por inúmeros servidores, alguns deles dedicados ao HMT e outros que servem o HMT em conjunto com outras aplicações. Na figura encontram-se também apresentados os protocolos de comunicação entre as diferentes aplicações.

Atualmente o HMT utiliza as diferentes bases de dados o acesso a informação. O PostgreSQL é utilizado como fonte de armazenamento de informação. No entanto, para um acesso mais eficiente a determinada informação, é utilizado o Elasticsearch através de uma aplicação que faz a gestão de acesso a esta base de dados, o Query Manager.

A Tabela 14 descreve com mais detalhe a infraestrutura da aplicação.

Tabela 14: Infraestrutura aplicacional

Máquina	Descrição
Browser	Neste servidor é onde se encontra alojada a aplicação gráfica da aplicação. Comunica com o ESP-API-V1 para o acesso ao conteúdo da aplicação.
ESP-API-V1	Esta máquina contém a aplicação HMT, que é responsável pela gestão dos conteúdos bem como a comunicação com outros sistemas e também o controlo de acesso do HMT. Comunica vários sistemas: o RabbitMQ <i>server</i> para a publicação de dados no Elasticsearch; <i>Query Manager</i> para ao acesso eficiente a informação; PostgreSQL para o acesso ao conteúdo aplicacional.
QUERY-MANAGER	Este servidor contém a aplicação que gere a comunicação com o Elasticsearch.
RabbitMQ	A arquitetura aplicacional utiliza uma <i>Message Queue</i> para a comunicação de outros sistemas. São aqui publicadas mensagens durante a utilização da aplicação (por exemplo, atualização de informação de pacotes).
PostgreSQL <i>server</i>	Base de dados relacional que armazena a toda a informação da aplicação. É nesta base de dados que está implementado o modelo de negócio do HMT.
ElasticSearch <i>server</i>	Base de dados destinada a pesquisa de informação de forma eficiente. Contém a parte do modelo de negócio do HMT necessária para responder rapidamente a inúmeras pesquisas (por exemplo, o número de pacotes que estão a aguardar a receção no armazém).
HMT-WEB-SERVICE	Neste servidor encontra-se alojada a aplicação responsável por consumir as mensagens publicadas no RabbitMQ, processá-las e publicá-las no Elasticsearch. No contexto do HMT, esta aplicação apenas consome as mensagens que são destinadas ao Elasticsearch.

Nas várias aplicações implementadas, o HMT e as restantes aplicações que o servem, encontram-se em utilização várias tecnologias. As principais são descritas na Tabela 15.

Tabela 15: Tecnologias utilizadas

Tecnologia	Descrição
Java	A tecnologia utilizada para o desenvolvimento da aplicação web, será o java na última versão disponibilizada – 1.8.
ElasticSearch	A bases de dados não relacional utilizada para neste projeto será o ElasticSearch na versão 2.3.3. Tendo em conta a linguagem utilizada para o desenvolvimento desta aplicação, será utilizada a java API disponibilizada, também na versão 2.3.3.
PostgreSQL	Como base de dados relacional será utilizado o PostgreSQL na versão 9.5.1. A ligação entre a aplicação java e a base de dados será feita através da <i>framework hibernate</i> na versão 5.1. Uma vez que serão tratados dados espaciais, será ainda utilizada a extensão PostGIS na versão 2.2
Maven	A ferramenta de <i>build</i> que será utilizada para este desenvolvimento será o Maven. Esta ferramenta será utilizada para o controlo de versões e dependências da aplicação.
RabbitMQ	A disponibilização de um meio de um meio de comunicação entre a aplicação aqui desenvolvida e os sistemas de gestão de encomendas será através de uma <i>Message Queue</i> . As aplicações externas deverão publicar mensagens (com a informação de encomendas) num formato específico, para que sejam consumidas e inseridas na base de dados. O RabbitMQ será utilizado para o armazenamento das mensagens provenientes de aplicações externas. A versão utilizada será 3.6.0. Será, assim, implementado um módulo ETL para que as mensagens sejam consumidas do RabbitMQ e publicadas na base de dados. Este módulo permitirá que as mensagens sejam manipuladas antes de serem inseridas na base de dados da aplicação. A possibilidade de manipulação, neste contexto, poderá ser importante para o ajuste de modelos de dados, etc.
AngularJS	O desenvolvimento da aplicação cliente será realizado em AngularJS. No que toca à versão, este detalhe não se encontra ainda definido uma vez que a versão utilizada na empresa onde o projeto será desenvolvido é a 1.0. Contudo, já se encontra disponível a versão 2.0 que é apresentada com melhorias no que toca ao desempenho

4.1.1 Controlo de acesso

O acesso ao HMT está condicionado aos colaboradores da Jumia Services. O controlo de sessões é realizado através do sistema de autenticação Google Sign-In. No entanto, para além do controlo de acesso à aplicação, o HMT faz também uma filtragem de acesso às funcionalidades. Esta filtragem é feita com recurso a um módulo de ACL (*Access Control List*).

O ACL desenvolvido na Jumia Services é composto por 4 áreas:

- Gestão de funções;
- Gestão de permissões de utilizadores;
- Filtragem de dados;

- Verificação de permissões.

Através deste módulo é possível especificar que, por exemplo, determinado utilizador tem acesso a um conjunto de funcionalidades num determinado armazém. Oferece também a possibilidade de definir perfis. Isto é, permite definir o acesso a um conjunto de funcionalidades e associá-lo a um perfil e, posteriormente, atribuir esse perfil a um ou mais utilizadores.

4.2 Requisitos

Qualquer *software* deve ser dotado de requisitos técnicos e funcionais que acrescentem valor ao produto. Neste sentido, a presente secção faz referência aos requisitos definidos para o desenvolvimento das novas funcionalidades na aplicação HMT.

A segunda fase do projeto descrito neste documento assentou no desenvolvimento de novas funcionalidades a um dos produtos da suite da Jumia Services. Estas funcionalidades devem apenas ser disponibilizadas a um conjunto restrito de utilizadores. A Figura 12 apresenta, sob a forma de um diagrama, os casos de uso estabelecidos para o desenvolvimento deste novo módulo que são detalhados nas subsecções que se seguem.

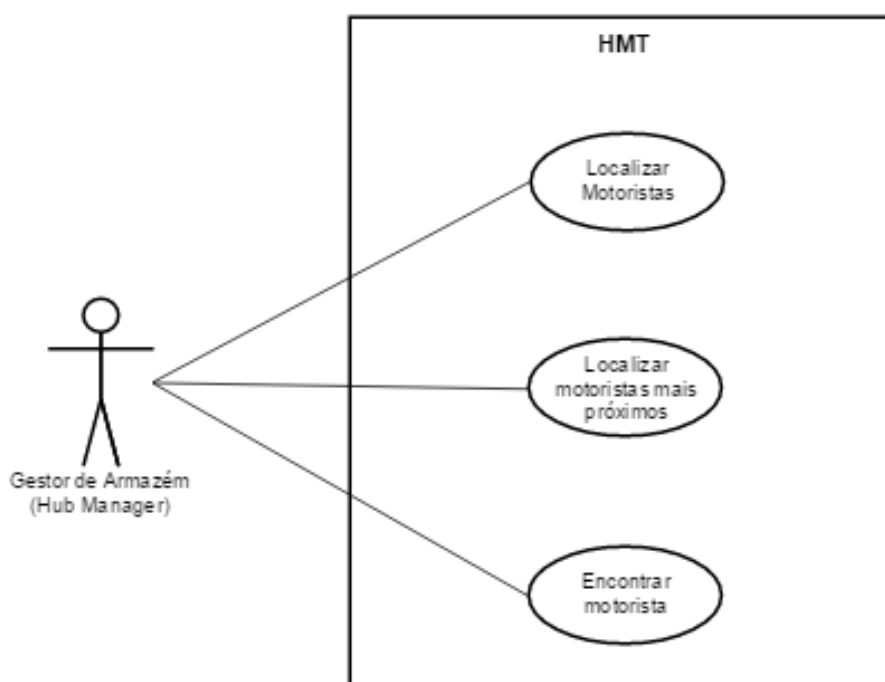


Figura 12: Casos de uso

As novas funcionalidades devem ser disponibilizadas numa área específica da aplicação e o seu acesso deve ser controlado. Desta forma, apenas utilizadores específicos devem ter acesso a este conjunto de funcionalidades.

Ainda no contexto das permissões dos utilizadores, importa referir que um utilizador não poderá visualizar motoristas associados a armazéns aos quais não tem acesso. Todos os motoristas estão associados a um armazém e o acesso dos utilizadores da aplicação é realizado ao nível do armazém.

A aplicação deverá garantir o acesso às localizações dos motoristas nas últimas 24 horas.

4.2.1 Localização de motoristas

O HMT deve disponibilizar, a todos os gestores de armazém um modo de aceder à localização de todos os seus motoristas. Assim, deve ser disponibilizada na aplicação uma nova área que disponibiliza a visualização, num mapa, de todos os motoristas do armazém em que o utilizador tem a sessão iniciada. A par com a localização dos utilizadores, este ecrã deve também disponibilizar informações relativamente ao motorista como o nome, email e o contacto telefónico. O utilizador deve ter ainda a capacidade de reajustar a região de visualização dos motoristas.

A Figura 13 apresenta o diagrama de atividades relacionadas com a funcionalidade aqui descrita.

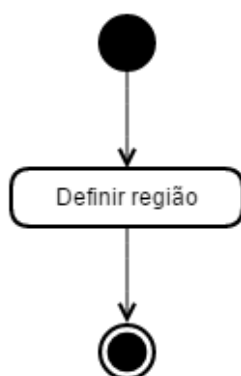


Figura 13: Diagrama de atividade - ver motoristas

4.2.2 Identificação de parceiros mais próximos

Esta funcionalidade pretende disponibilizar aos utilizadores a capacidade de encontrar os utilizadores dentro de um determinado raio. De forma semelhante à funcionalidade anterior, a par com a visualização no mapa deve constar a listagem dos motoristas que se encontram nas condições definidas. Os resultados devem ser apresentados ordenados pela distância a que os motoristas se encontram.

No caso de não serem encontrados resultados, essa informação deve ser apresentada ao utilizador. Na Figura 14 é apresentado o diagrama de atividades com os passos necessários para a localização dos motoristas mais próximos.

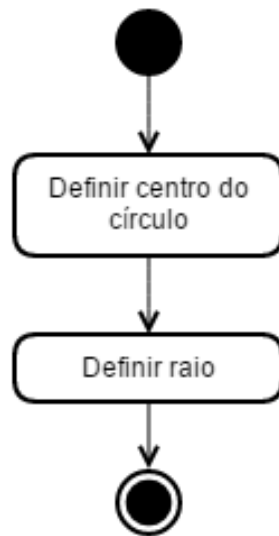


Figura 14: Diagrama de atividade - motoristas mais próximos

4.2.3 Encontrar motorista

O HMT deve disponibilizar aos utilizadores a capacidade de, através de texto livre, encontrar os seus motoristas. Assim, nesta área da aplicação o utilizador deverá ter a possibilidade de encontrar um motorista através do nome, contacto telefónico ou email. Do resultado da pesquisa, o utilizador deverá escolher um motorista que será visualizado no mapa. No caso de não ser encontrado qualquer resultado, deve ser apresentada uma mensagem ao utilizador com essa informação.

A Figura 15 apresenta o fluxo de atividades correspondentes à pesquisa e localização de um motorista.

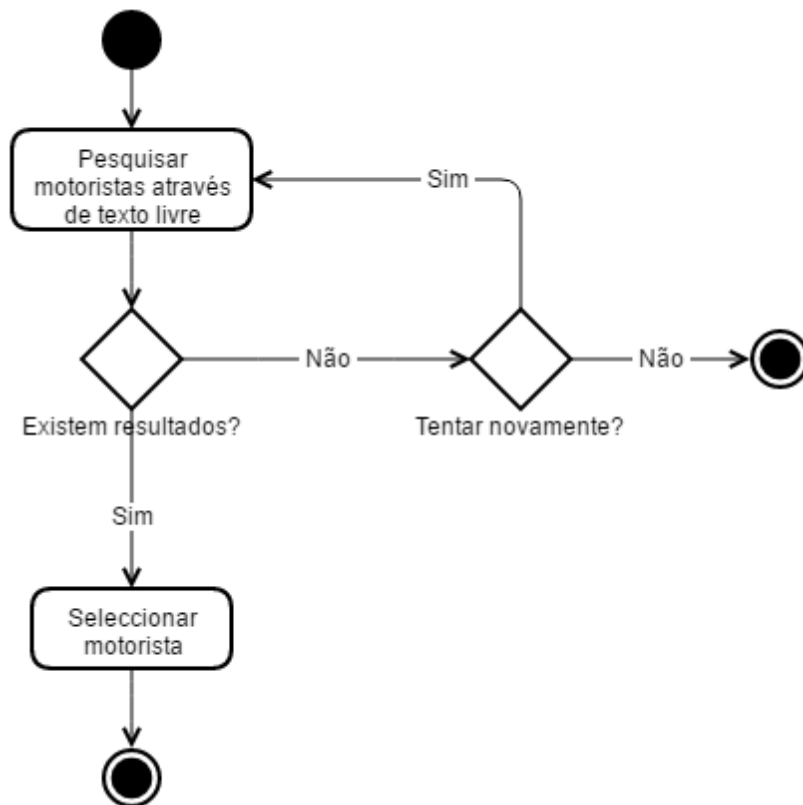


Figura 15: Diagrama de atividades - pesquisa de motoristas

4.3 Design

Dada a complexidade envolvida no desenvolvimento do novo módulo da aplicação, esta secção encontra-se dividida em múltiplas subsecções com o objetivo de especificar os vários componentes envolvidos.

4.3.1 Armazenamento de dados

Atendendo às características de cada uma das bases de dados analisadas e, com base nos resultados dos testes realizados, optou-se pela utilização do Elasticsearch para o desenvolvimento do protótipo. Apesar dos fracos resultados na inserção de informação, quando comparados com os resultados do PostgreSQL, o Elasticsearch surge como uma boa escolha no que toca à pesquisa de informação.

Assim, numa aplicação em que o foco será a velocidade com que os resultados são apresentados ao utilizador, optou-se pela utilização da base de dados NoSQL. Contudo, é também realizado o armazenamento da informação na base de dados PostgreSQL com o objetivo de servir, no futuro, de fonte de acesso ao histórico de dados dos motoristas.

Sendo a solução de pesquisa de informação implementada numa base de dados não relacional, o modelo de dados desenvolvido de acordo com as características da mesma. A Figura 16 apresenta o modelo de dados criado para responder aos requisitos especificados.

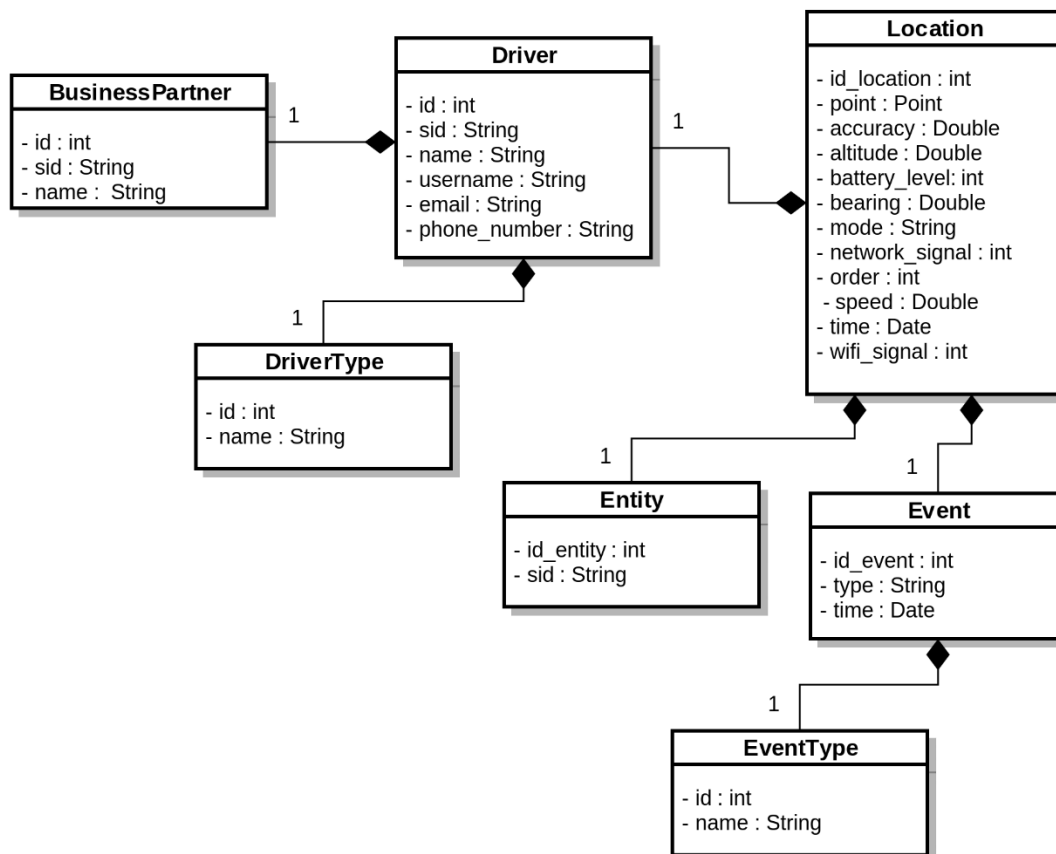


Figura 16: Modelo de dados

No modelo de dados desenhado identificamos as seguintes entidades:

- **BusinessPartner** – Representa os armazéns da Jumia Services;
- **Driver** – Representa os motoristas da Jumia Services. Este encontra-se relacionado com o armazém em que opera;
- **DriverType** – Identifica o tipo de motorista;
- **Location** – Representa cada uma das localizações registadas pelos motoristas durante o seu percurso;
- **Entity** – Representa os pacotes entregues pelos motoristas. Cada localização pode conter um conjunto de entidades, que são as que os motoristas se encontram a processar;
- **Event** – Representa os eventos processados pelos motoristas;
- **EventType** – Define o tipo de evento que pode ser a entrega de um pacote, pausa para o almoço, entre outros.

Uma vez que a implementações dos requisitos é feita sobre uma base de dados não relacional, os modelo de dados definido e apresentado na Figura 16 foi desnormalizado. A representação deste modelo é feita no formato JSON, sendo também este o formato com que os documentos são inseridos no Elasticsearch.

```
{
  "id": "long",
  "point": "Point",
  "accuracy": "double",
  "altitude": "double",
  "batteryLevel": "long",
  "bearing": "double",
  "mode": "String",
  "networkSignal": "long",
  "speed": "double",
  "time": "DateTime",
  "wifiSignal": "long",
  "driver": {
    "id": "long",
    "sid": "String",
    "name": "String",
    "username": "String",
    "phoneNumber": "String",
    "email": "String",
    "businessPartner": {
      "id": "long",
      "sid": "String",
      "name": "String"
    },
    "type": {
      "id": "String",
      "name": "String"
    }
  },
  "entities": [
    {
      "id": "long",
      "sid": "String",
      "trackingNumber": "String"
    }
  ],
  "event": {
    "id": "long",
    "time": "DateTime",
    "type": {
      "id": "long",
      "name": "String"
    }
  }
}
```

Figura 17: Elasticsearch *schema*

No desenvolvimento de *software*, ligado ao modelo de dados surge o modelo de classes. No caso do HMT, foram desenvolvidos dois modelos de classes. O primeiro é o utilizado pela aplicação para a gestão de dados e o segundo constitui a resposta de dados para a camada de

interface gráfica. Os diagramas de classes apresentados na Figura 18 e Figura 19 são correspondentes ao diagrama de classes utilizado para modelação do negócio e o diagrama de classes utilizado na resposta para a camada gráfica da aplicação.

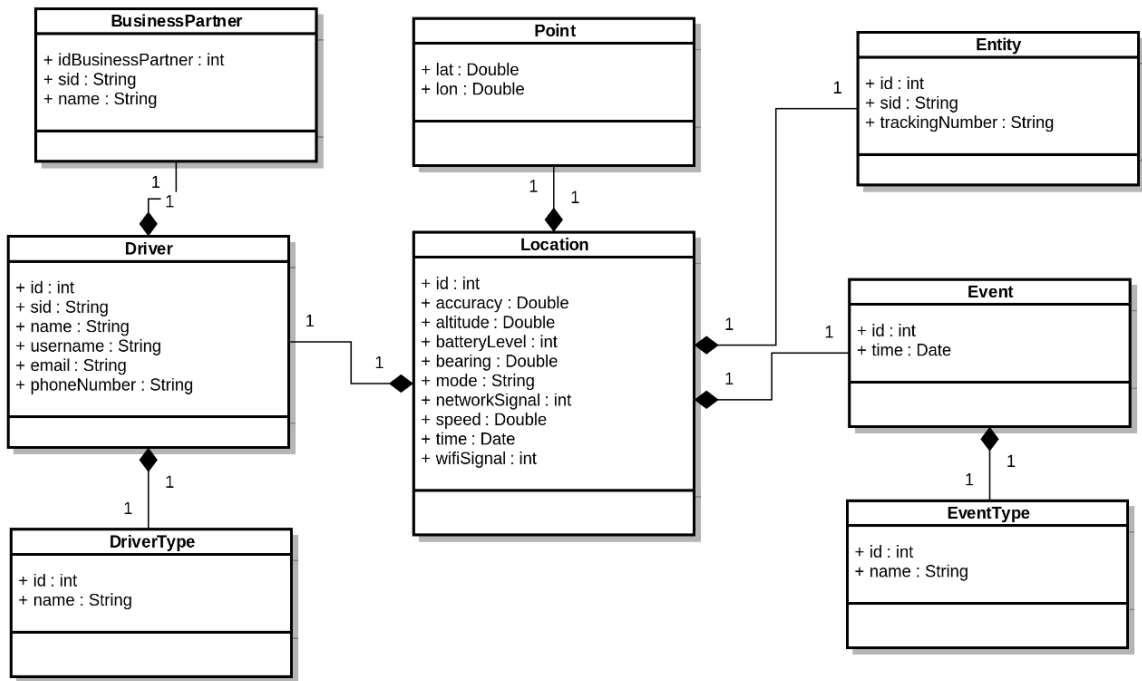


Figura 18: Diagrama de classes – negócio

As diferenças entre os dois modelos baseiam-se, sobretudo, na redução do número de propriedades das classes.

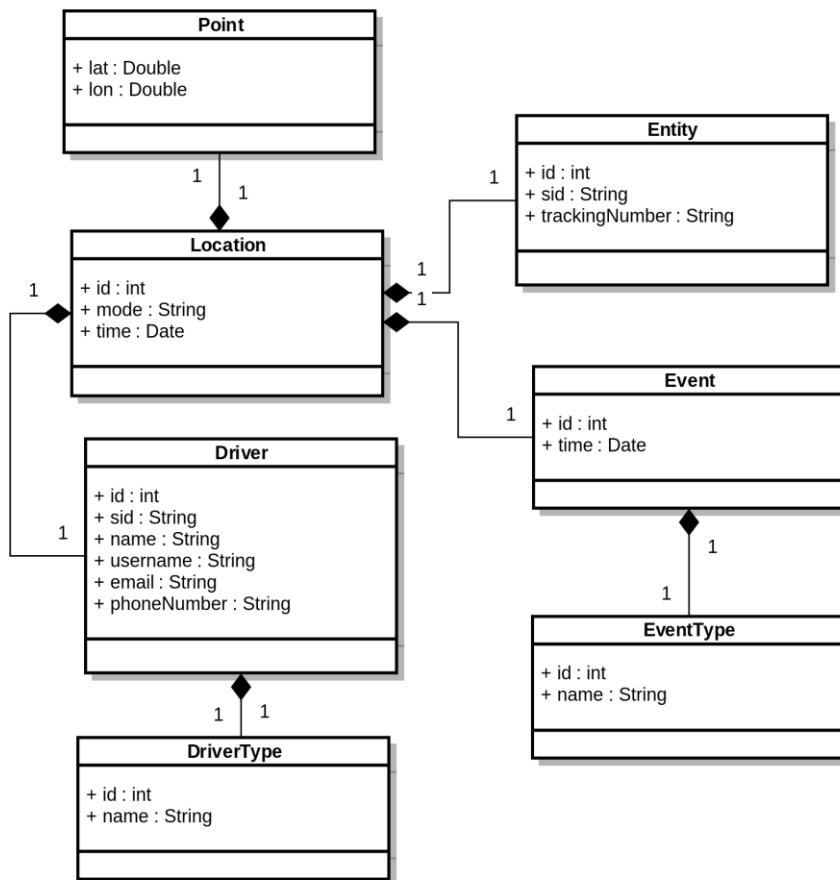


Figura 19: Diagrama de classes - UI

4.3.2 Publicação de dados

Conforme mencionado nos objetivos deste projeto (secção 1.3.2), está fora do âmbito deste desenvolvimento, a captura e publicação dos dados recolhidos pela D-aap. Assim, no âmbito to projeto aqui descrito, foi criado um meio de comunicação com o recurso ao RabbitMQ.

Para popular o Elasticsearch com a informação proveniente da utilização da D-app, foi criada um novo canal no RabbitMQ que permite que a D-app publique dados para serem consumidos pelo Routing-manager e registados no Elasticsearch.

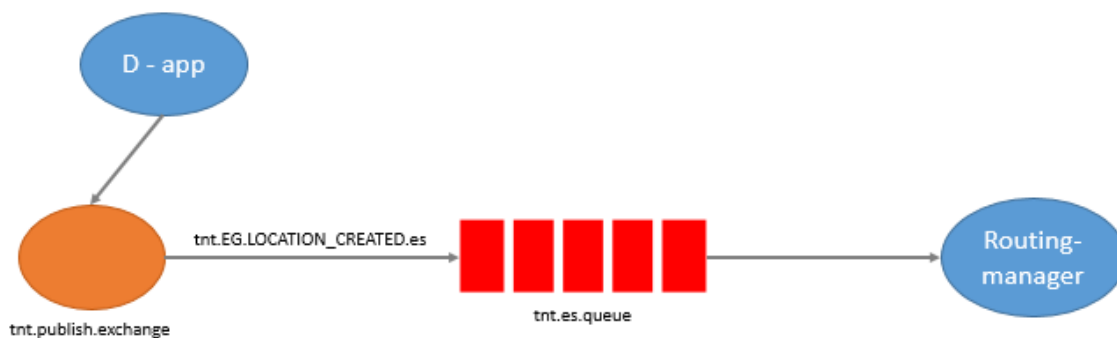


Figura 20: Fluxo de dados no RabbitMQ

A Figura 20 descreve o fluxo de mensagens no RabbitMQ, desde a sua partilha por parte da D-app até ao consumo realizado pelo routing-manager. O processo de leitura das mensagens publicadas no RabbitMQ e a sua publicação é realizado pelo routing-manager, um módulo já desenvolvido, que apenas requer a configuração das novas listas de mensagens.

4.3.3 Acesso aos dados

Para o acesso aos dados, o Elasticsearch disponibiliza uma linguagem de pesquisa – QueryDSL – baseada em JSON. Para este projeto, em vez de construir as *queries* em tempo de execução, estas foram pré-registadas na base de dados, o que oferece as seguintes vantagens:

- Facilidade de alteração das pesquisas sem a necessidade de recompilar a aplicação e realizar uma nova instalação, o que obrigaria a que aplicação ficasse indisponível durante algum tempo.
- Melhoria do desempenho na medida em que a aplicação não necessita de reconstruir a query a cada novo pedido.

As pesquisas pré-registadas, no Elasticsearch, designam-se *search templates*. Cada um dos *search templates* é identificado por um ID único. Para servir os requisitos delineados para este projeto, foram criadas as pesquisas apresentadas na Tabela 16.

Tabela 16: Search Templates

Id	Descrição
hmt_driversLocations	Localiza todos os motoristas de um ou mais armazéns. Como parâmetros recebe apenas os identificadores dos armazéns.
hmt_searchDriver	Permite a pesquisa de motoristas através de texto livre. O texto é aplicado a múltiplas propriedades do motorista como o nome, email e telefone. Nesta pesquisa, os parâmetros recebidos são o texto a aplicar na pesquisa e a lista dos identificadores dos armazéns nos quais a pesquisa deve incidir.

hmt_nearDrivers	Localiza todos os motoristas que se encontram dentro de um raio definido. Os parâmetros são o centro da circunferência, a distância máxima dos motoristas e a lista dos armazéns.
-----------------	---

O acesso ao ElasticSearch para a realização de pesquisas por parte do HMT é feito com o recurso a uma aplicação designada query-manager. Uma vez que o ElasticSearch contém a informação de todos os motoristas de todos os armazéns, a forma de controlar o acesso aos motoristas passa pela utilização dos identificadores dos armazéns em todas as pesquisas.

4.3.4 Serviços

A comunicação entre a camada de interface gráfica e a aplicação sediada remotamente é feita através de pedidos REST. A aplicação HMT tem um padrão definido para a criação de novos *endpoints* de acesso a dados. Todos os serviços caracterizam-se por serem compostos pelos identificadores da rede de armazéns e do armazém em que o utilizador tem a sua sessão iniciada Assim, o formato dos serviços é construído da seguinte forma:

- Identificador da API: esp-api-v1
- Identificador da rede de armazéns: /sp/<identificador>
- Identificador do armazém: /bp/<identificador>
- Entidade processada pela *endpoint*, por exemplo, pacotes: /packages
- Múltiplas funções podem ser adicionadas depois da especificação da entidade, por exemplo, para a exportação: /export

As regras de definição de *endpoints* foram tidas em conta aquando da criação dos serviços para o acesso à informação dos motoristas. A Tabela 17 apresenta os serviços criados.

Tabela 17: Serviços disponibilizados pelo HMT

Serviço	Caminho
Localização de motoristas	/esp-api-v1/sp/<spId>/bp/<bpId>/drivers/locations
Motoristas mais próximos	/esp-api-v1/sp/<spId>/bp/<bpId>/drivers/nearest
Encontrar motorista	/esp-api-v1/sp/<spId>/bp/<bpId>/drivers

Considerando a possibilidade de o número de motoristas ser elevado, todos os serviços criados retornam um número máximo de resultados (1000). Desta forma, os resultados são disponibilizados sob a forma de páginas.

A implementação dos serviços é realizada sob a forma de camadas com o objetivo de que cada uma possa ser reaproveitada em contextos diferentes. Para cada um dos novos serviços criados, foram desenhados os diagramas de sequência que representam a implementação dos mesmos. Os diagramas estão representados na Figura 21, Figura 22 e Figura 23.

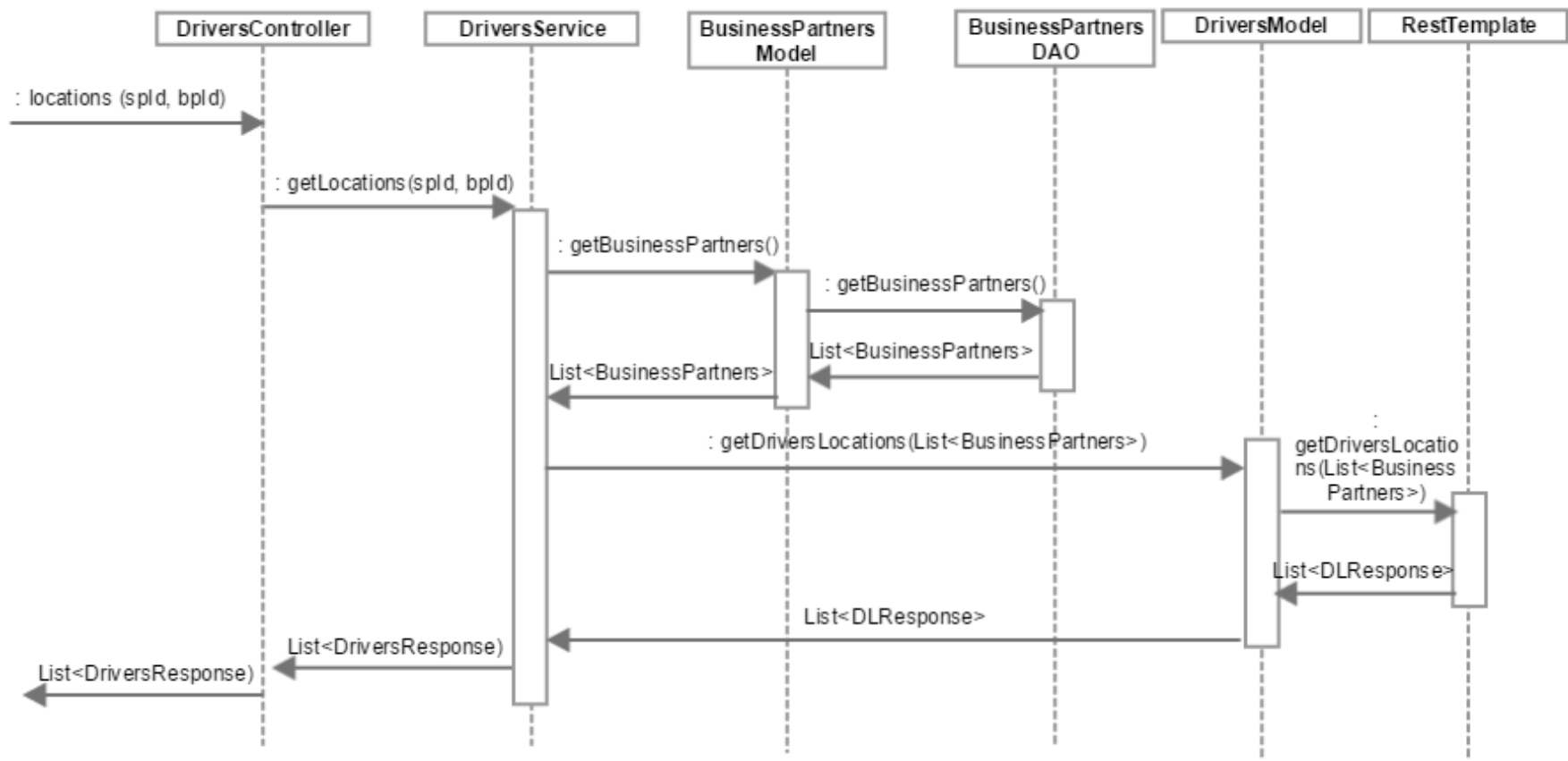


Figura 21: Diagrama de Sequência - Localização de motoristas

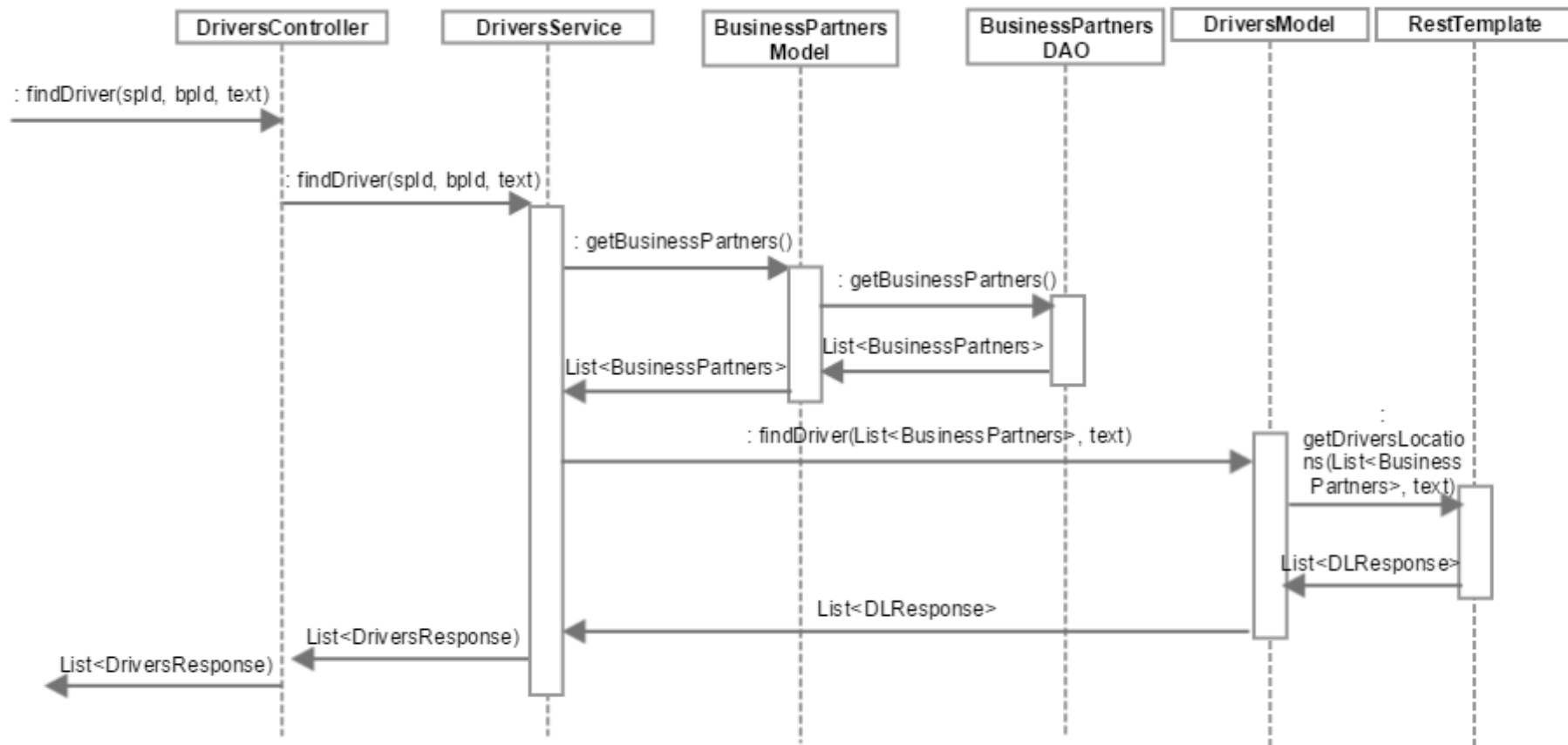


Figura 22: Diagrama de Sequência - Encontrar motorista

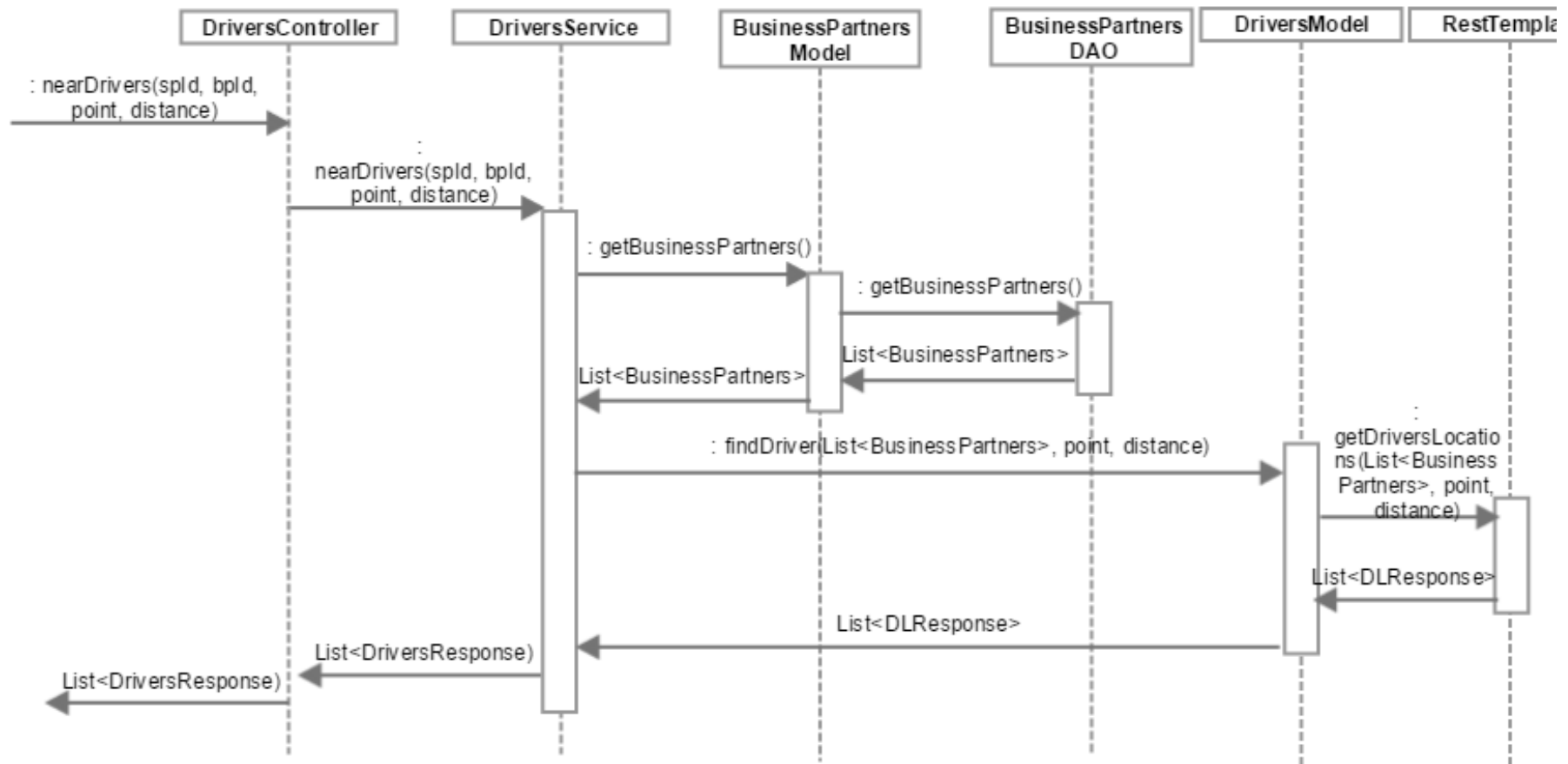


Figura 23: Diagrama de Sequência - Motoristas mais próximos

Nos diagramas verificamos a existência várias classes com funções bem definidas, que são caracterizadas da seguinte forma:

- **Controller** – Expõe o serviço na API. Esta classe implementa as validações das chamadas ao serviço. Por exemplo, confirma se todos os parâmetros necessários à execução do serviço estão definidos;
- **Service** – Implementa as regras de negócio;
- **Model** – define o acesso à informação. No exemplo apresentado, para o acesso aos motoristas utiliza uma interface com a base de dados. No entanto, para o acesso à localização dos mesmos, recorre a uma interface de serviços REST para comunicar com o query manager que acede ao Elasticsearch;
- **DAO** – Definem o acesso à base de dados;
- **Template** - Implementam a comunicação com outros serviços REST.

4.3.5 Camada de interação com o utilizador

A interface gráfica de uma aplicação tem um papel importante na satisfação com que os utilizadores a utilizam, o que faz com que esta seja uma área desenvolvida com especial cuidado, nomeadamente no que diz respeito à interação do utilizador.

Tal como mencionado anteriormente neste relatório, o HMT é desenvolvido em AngularJS. Para uniformizar a aplicação, os novos desenvolvimentos, de forma semelhante ao que foi feito na implementação dos serviços, foram implementados para que a maior parte do código possa ser reaproveitado em requisitos futuros.

Com o objetivo de apresentar a forma como a camada gráfica se encontra organizada é apresentada abaixo a Figura 24. Como podemos ver pela figura, existem vários módulos envolvidos no desenvolvimento desta camada de software:

- **View** – Camada onde são implementados os componentes gráficos da aplicação.
- **Controller** – Camada onde são implementadas as funções de controlo dos componentes gráficos.
- **Service** – Implementa os serviços de comunicação com a API para obtenção de informação.
- **Constants** – Define um conjunto de constantes utilizadas ao longo do novo módulo.
- **Utilities** – define um conjunto de funções genéricas que podem ser reaproveitadas ao longo da aplicação.

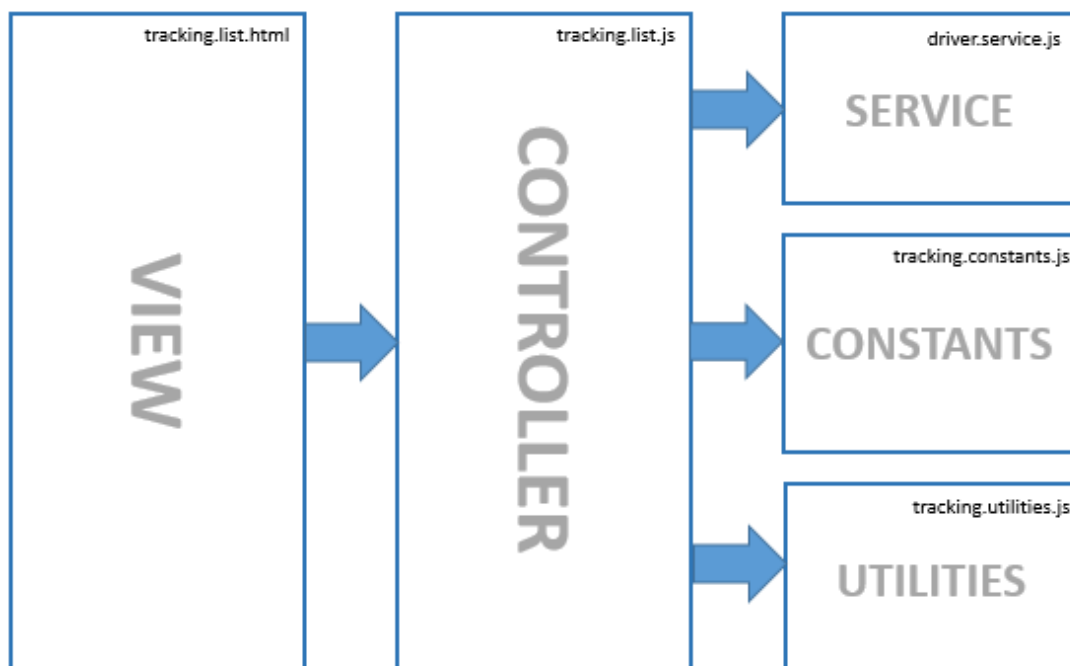


Figura 24: Organização da interface gráfica

4.4 Desenvolvimento

Nesta secção do relatório são apresentadas as etapas do desenvolvimento do novo módulo no HMT. Durante o desenvolvimento deste novo módulo na aplicação, foram tidas em conta as melhores práticas para o desenvolvimento de *software*.

4.4.1 Pesquisas

Conforme mencionado na secção 4.3.3 deste documento, as pesquisas para o acesso à informação foram registadas como *search templates* no Elasticsearch. Foram assim criadas 3 pesquisas que respondem às necessidades da aplicação, que se baseiam no acesso à localização atual dos motoristas, a pesquisa de motoristas e a identificação dos motoristas mais próximos.

As pesquisas realizadas ao Elasticsearch dividem-se em dois blocos principais:

- **Filtros** – É neste bloco da query que são definidos os filtros que são ser aplicados no universo de dados a que a pesquisa se aplica.
- **Agregações** – Permite agrupar os resultados por um ou mais elementos.

Para além destes, é possível ainda definir um outro bloco com filtros a ser aplicados depois das agregações. Contudo, este não foi utilizado nos *templates* criados para a aplicação. No caso particular das pesquisas desta aplicação, foram aplicados os seguintes filtros:

- **Armazéns** – excluem todos os motoristas que não pertencem à lista de armazéns à qual o utilizador tem acesso. Este é um filtro utilizado em todas as pesquisas, uma vez que garante que os utilizadores não acedem indevidamente às localizações de todos os motoristas da empresa;
- **Distância** – Este filtro é aplicado no apenas na pesquisa dos motoristas mais próximos de um determinado ponto;
- **Texto** – Utilizado na pesquisa de motoristas, serve para encontram os colaboradores cujos dados identificativos (nome, email e telefone) fazem correspondência com o texto introduzido pelo utilizador na aplicação.

No que toca às agregações, foi utilizada uma comum a todas as pesquisas:

- **Últimas localizações** – Agrega os resultados, depois de aplicados os filtros, por motorista através do seu identificador e devolve a sua última localização.

A Figura 25 apresenta o exemplo de uma das pesquisas criadas. Neste caso, para a obtenção última localização de todos os motoristas dentro de um determinado raio. Esta pesquisa recebe por parâmetro uma lista de identificadores de armazéns e um ponto (latitude e longitude) que define o centro a utilizar na pesquisa.

```

{
  "from": 0,
  "size": 0,
  "query": {
    "filtered": {
      "filter": {
        "and": [
          {
            "terms": {
              "driver.businessPartner.sid.raw": [
                "{{bpID}}",
                "{{.}}",
                "{{bpID}}"
              ]
            }
          },
          {
            "geo_distance": {
              "distance": "1km",
              "point": {
                "lat": "{{lat}}",
                "lon": "{{lon}}"
              }
            }
          }
        ]
      }
    },
    "aggs": {
      "drivers": {
        "terms": {
          "field": "driver.sid.raw"
        },
        "aggs": {
          "top_locations": {
            "top_hits": {
              "sort": [
                {
                  "time": {
                    "order": "desc"
                  }
                }
              ]
            }
          }
        }
      }
    }
  }
}

```

Figura 25: Pesquisa - motoristas mais próximos

4.4.2 Camada de negócio

Durante todo o desenvolvimento deste projeto, foi tida em consideração a aplicação das melhores práticas do desenvolvimento de *software*, por exemplo, com a implementação de padrões de design de software.

Nesta camada foram criadas, para além do modelo de dados, classes para a representação dos dados esperados nos serviços. A classe representada na Figura 26 é partilhada por todos os serviços.

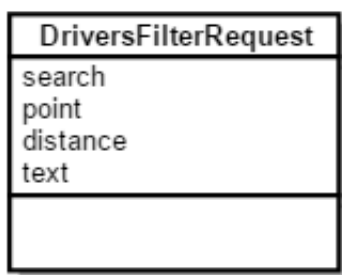


Figura 26: Classe para definição dos pedidos

Com base na propriedade *search* é definida a pesquisa que o utilizador pretende realizar e os restantes parâmetros servem para complementar a pesquisa. O identificador do armazém utilizador na pesquisa é aquele que se encontra definido no URL do serviço, conforme definido na Tabela 17.

Com base nos parâmetros recebidos nos pedidos aos serviços é construído o *payload* do pedido para o query manager, que define inúmeros parâmetros:

- Identificador da pesquisa a realizar;
- Parâmetros a serem aplicados na pesquisa;
- Tipo de resposta (Estatísticas ou dados). Para os casos de uso apresentados, pretende-se que sejam retornados os dados resultantes das pesquisas;

Com base nesta informação, o query manager executa uma pesquisa no Elasticsearch e devolve os resultados dessa pesquisa no formato JSON e, com base nesses resultados, é feita a desserialização da resposta para o modelo de dados do HMT. Esse modelo é posteriormente mapeado para o modelo que define a resposta dos serviços da aplicação.

Durante todo este processo são utilizadas diferentes tipos de classes para funções muito específicas:

- **Builders** para a criação de novos objetos;
- **Transformers** para proceder ao mapeamento de classes de um modelo origem para classes de um modelo destino.

Os serviços disponibilizados pelo HMT foram utilizados os códigos HTTP para definir o sucesso/insucesso dos pedidos realizados. A Tabela 18 faz uma apresentação dos códigos utilizados.

Tabela 18: Códigos de resposta HTTP utilizados

Código	Descrição
200	Pedido realizado com sucesso
401	Tentativa de acesso a um recurso não autorizado
500	Erro ocorrido durante a execução do pedido

4.4.3 Interface gráfica

O AngularJS disponibiliza a possibilidade da criação de módulos, que não são mais do que formas de agrupar as diferentes partes da aplicação. Assim, para a criação do novo módulo do HMT, que permite o acesso à localização dos motoristas, foi criado um novo módulo - *tracking*.

Para as 3 funcionalidades solicitadas foram criados apenas 2 novos ecrãs na aplicação. A localização e a pesquisa de motoristas são realizadas no mesmo ecrã, com o objetivo de reduzir a necessidade do utilizador alterar de menu para efetuar operações semelhantes. Quanto à localização dos motoristas mais próximos, esta funcionalidade é acedida num ecrã diferente da aplicação.

A inserção de mapas na aplicação do HMT é conseguida através de uma biblioteca *open source*, designada *ngmap*, que disponibiliza um vasto leque de funções que foram incorporadas na aplicação, como por exemplo, a identificação de motoristas no mapa através de marcadores. Foi então adicionada ao projeto a dependência do *ngmap* na versão 1.17.7.

O módulo de *tracking*, no HMT envolveu o desenvolvimento de vários componentes utilizados ao longo da sua definição:

- **Controller** – Este é o componente responsável pelo controlo dos componentes existentes no ecrã bem como pela comunicação entre a aplicação e os serviços;
- **Services** – Foi criado um serviço que disponibiliza as funções necessárias para a comunicação com os serviços disponibilizados pelo HMT.
- **Utilities** – Este componente disponibiliza um conjunto de funções auxiliares para a ajuda ao processamento de dados. Um exemplo de uma função disponibilizada por este componente é a função para a conversão das localizações dos utilizadores em dados compatíveis com o *ngmap*.

Para além da organização em componentes para que possam ser reutilizados, também os ecrãs foram criados e divididos em vários componentes que pudessem ser partilhados. Alguns exemplos dos componentes gráficos criados são:

- Diretiva para encapsular a comunicação com os mapas. Uma vez que no contexto deste projeto foram criados dois ecrãs com um mapa cada, esta diretiva é partilhada;
- Filtro para a apresentação do tempo desde a última ligação do motorista com o servidor. Com base nos dados recebidos do servidor, este filtro é responsável por interpretar a data da última comunicação e apresentar a diferença em segundos ou minutos;
- Filtro para a apresentação colorida do modo de entrega em que o motorista se encontra. Por exemplo, modos EMERGENCY, RUN e LOW são apresentados em vermelho, amarelo e verde respetivamente;
- O componente de filtros, existente no ecrã de pesquisa dos motoristas mais próximos, foi encapsulado numa diretiva. Embora esteja apenas a ser utilizado neste ecrã, é um componente que pode facilmente vir a ser utilizado noutros, mesmo que com filtros diferentes. A diretiva foi criada de modo a que seja dinâmica e que possam ser adicionados novos filtros facilmente.

As próximas secções (4.4.3.1, 4.4.3.2 e 4.4.3.3) fazem a apresentação dos ecrãs criados para a disponibilização das funcionalidades solicitadas.

4.4.3.1 *Localização de motoristas*

A aceder ao menu lateral de tracking, o ecrã apresentado na Figura 27 é vista por defeito da qual o utilizador dispõe. Através deste ecrã o utilizador tem a possibilidade de encontrar todos os motoristas que se encontram dentro da janela definida no mapa. A par com o mapa encontra-se a lista de todos os motoristas associados ao armazém no qual o utilizador tem a sessão iniciada.

Nesta lista de motoristas são disponibilizadas as seguintes informações relativas a cada um dos motoristas:

- Nome;
- Modo em que o motorista se encontra a emitir o sinal (Low mode, Run mode, Emergency mode);
- Data da última emissão do sinal por pelo motorista.

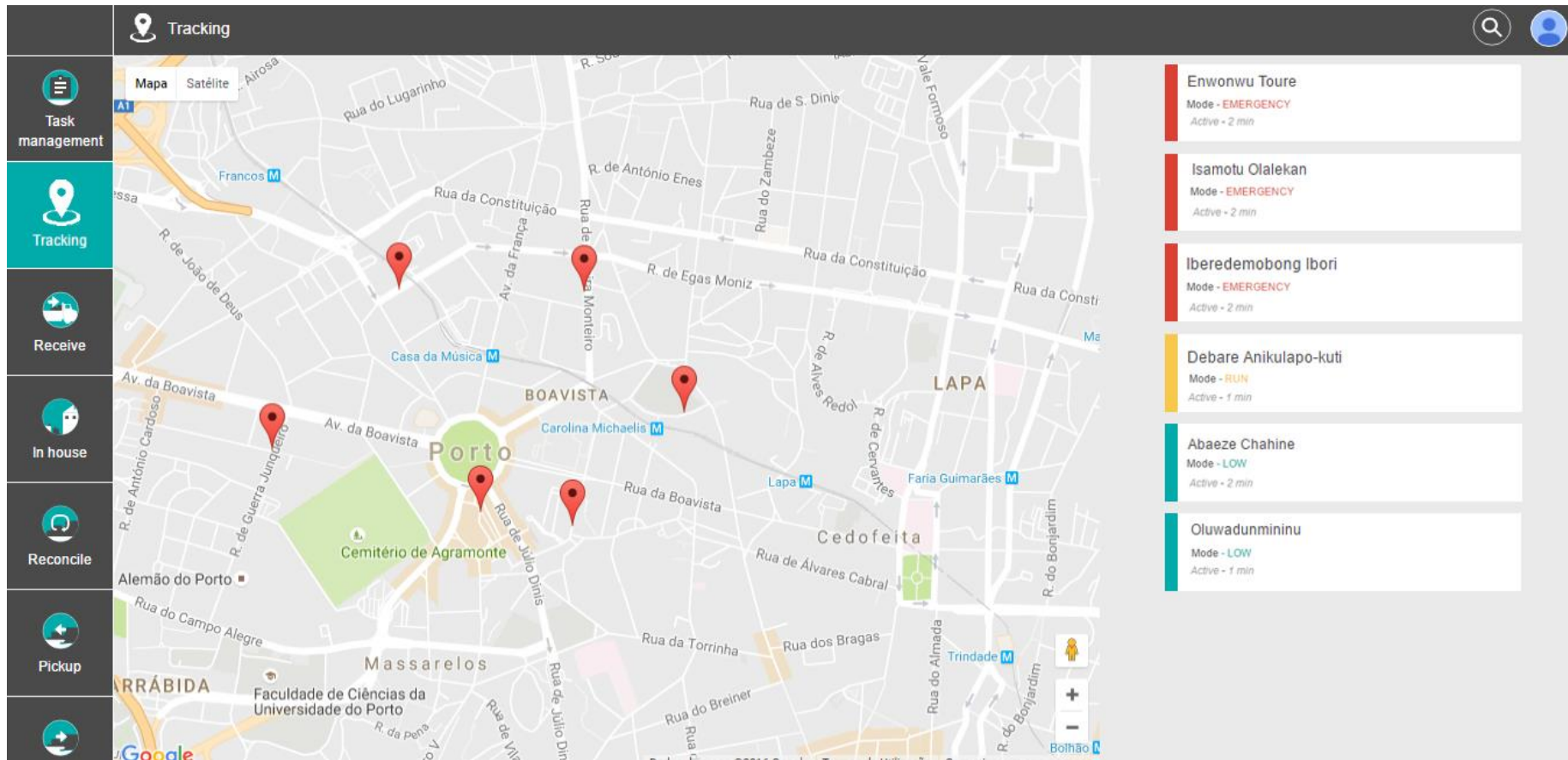


Figura 27: Localização de motoristas

4.4.3.2 *Encontrar motorista*

Na mesma vista apresentada pela Figura 27 o utilizador tem a possibilidade por pesquisar um motorista em específico. Para isso deve aceder ao ícone de pesquisa no canto superior direito, como é sugerido na Figura 28.

De forma semelhante à funcionalidade anterior, os resultados da pesquisa são apresentados ao lado direito do mapa, sendo também apresentadas as mesmas informações relativas a cada motorista do resultado da pesquisa.

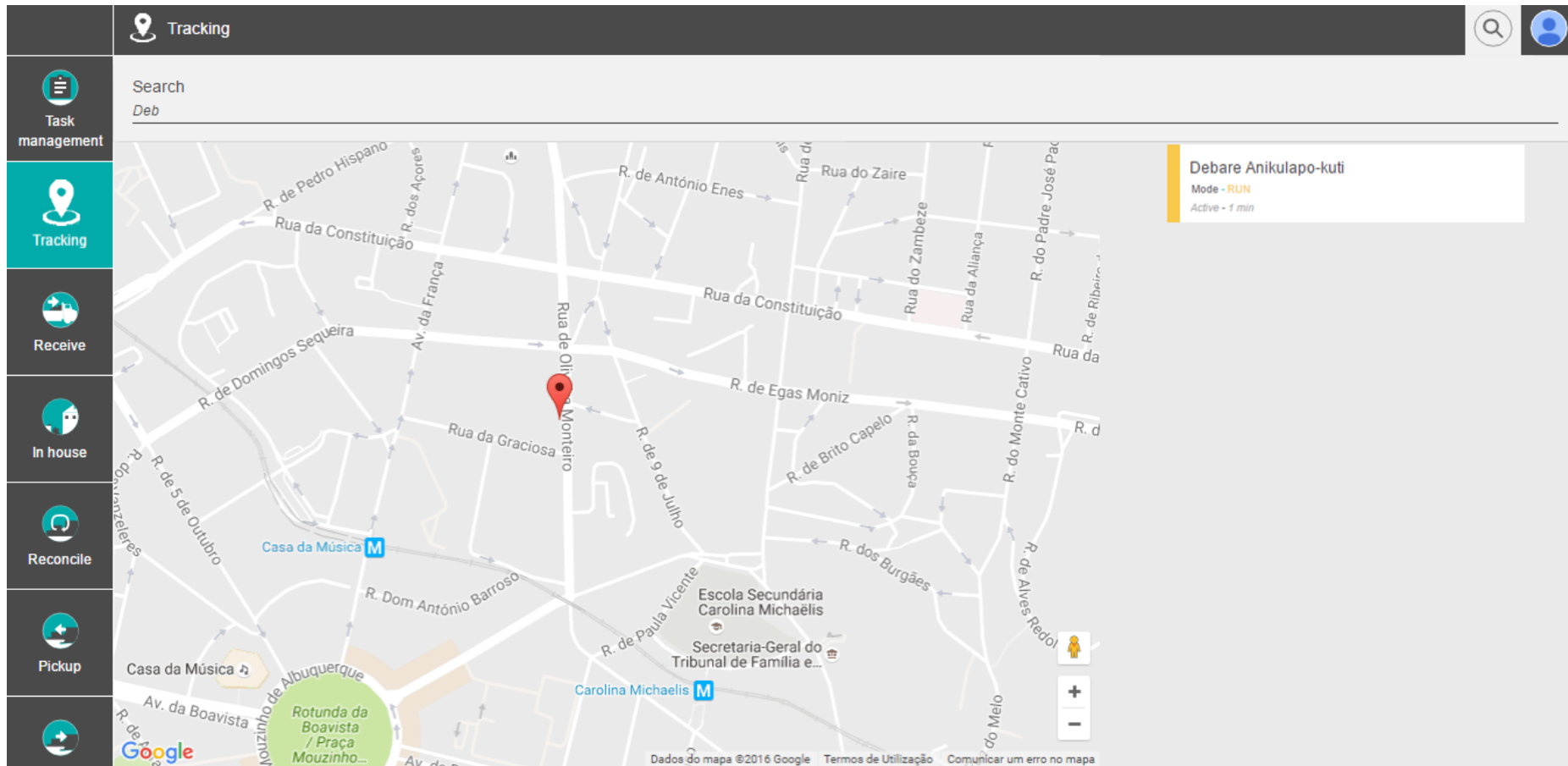


Figura 28: Encontrar motorista

4.4.3.3 *Identificação de motoristas*

Para a identificação dos motoristas mais próximos o utilizador deverá utilizar um novo ecrã que disponibiliza um conjunto específico de filtros. Este ecrã é apresentado na Figura 29. No conjunto de filtros o utilizador deverá definir o ponto que especifica o centro da circunferência na qual pretende realizar a pesquisa e o raio.

De forma semelhante os restantes ecrãs, os resultados são apresentados à direita do mapa.

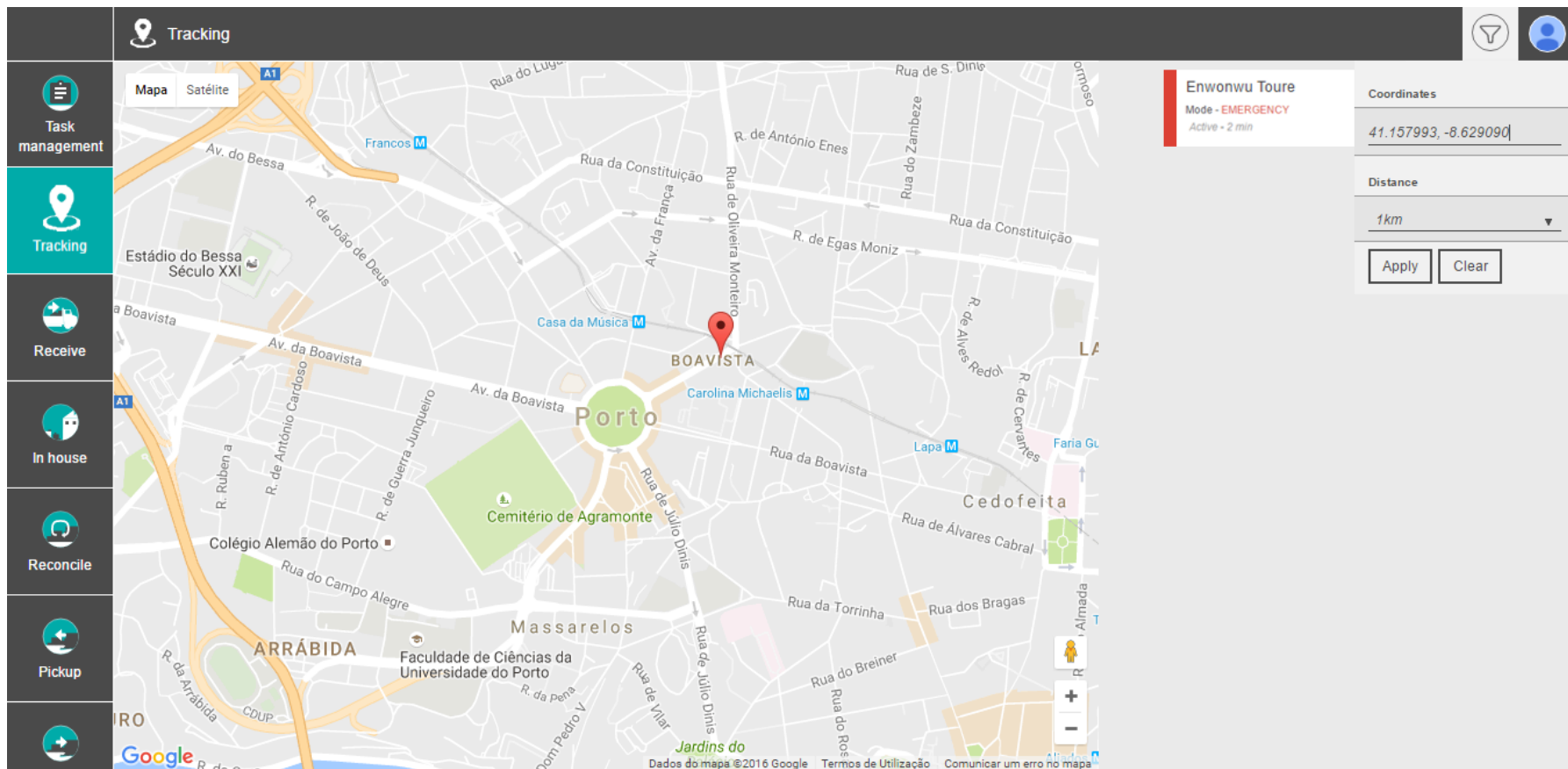


Figura 29: Motoristas mais próximos

4.5 Avaliação

A qualidade de um produto é ditada não só pelo valor que este trás às empresas que o utilizam, mas também pela aceitação que os utilizadores fazem do mesmo. Neste sentido, foi realizado um inquérito de satisfação com o objetivo de avaliar a qualidade do produto desenvolvido e também de apurar eventuais melhorias a serem realizadas.

O inquérito realizado (apresentado anexo 1) inclui um pequeno grupo de questões acerca da usabilidade daquilo que foi desenvolvido neste projeto e questões relativas à utilidade da informação providenciada pelo novo módulo criado no HMT.

Até à data de entrega deste relatório, apenas 15 colaboradores responderam ao inquérito realizado. Entre os inquiridos estão os gestores de armazém da Jumia Services e gestores de produto.

As questões colocadas aos utilizadores durante o inquérito, bem como as respostas obtidas são apresentadas nas secções seguintes (4.5.1 a 4.5.6).

4.5.1 Quão amigável classifica a interface da aplicação?

Para esta questão, que pretende obter a opinião dos utilizadores no que toca à sua satisfação relativamente à fácil utilização das funcionalidades, estavam disponíveis 5 opções de resposta (desde não amigável a extremamente amigável).

Num conjunto de 15 respostas obtidas, cerca de 57% dos utilizadores classificaram a usabilidade das novas funcionalidades como muito amigável. Sendo que os restantes a classificaram como muito amigável.

4.5.2 Está satisfeito com as funcionalidades disponibilizadas?

Nesta questão pretende-se determinar a satisfação, de um modo global, dos utilizadores no que toca às novas funcionalidades acrescentadas à aplicação. Todos inquiridos classificaram de positiva a satisfação com o novo módulo acrescentado ao HMT.

4.5.3 A informação disponibilizada é útil no seu dia-a-dia?

Esta é a primeira pergunta do inquérito que pretende determinar se as funcionalidades disponibilizadas na aplicação são uteis no dia a dia de quem as utiliza. Esta questão foi respondida de forma positiva por 69% dos utilizadores, tendo o restante público determinado como não aplicável.

Calcula-se que os utilizadores que responderem “Não aplicável” a esta questão, não estão relacionados com o trabalho realizado no armazém.

4.5.4 Acha as novas funcionalidades úteis para a melhoria do desempenho do seu armazém?

Nesta questão, que pretende determinar a utilidade das funcionalidades para a melhoria do desempenho da atividade nos armazéns, cerca de 31% dos inquiridos responderam “Não aplicável”. Dos restantes inquiridos, 8% classificaram as funcionalidades como não úteis enquanto que os restantes responderam de forma positiva.

4.5.5 Como responsável de armazém, com que frequência espera utilizar as novas funcionalidades?

A esta questão, que tem o objetivo de apurar a frequência com que os utilizadores irão utilizar as novas funcionalidades desenvolvidas na aplicação, 8% responderam que utilizariam 1 vez por dia. Dos restantes inquiridos, 31% responderam que utilizariam 2 vezes por dia e os restantes escolheram a opção “Não aplicável”.

4.5.6 Sugestões

A última secção do questionário deu a possibilidade aos utilizadores de deixarem sugestões de melhoria às novas funcionalidades da aplicação. Pretende-se que essas melhorias sejam tidas em conta em desenvolvimentos futuros. A lista de melhorias sugeridas pelos utilizadores é apresentada abaixo.

- O conjunto de funcionalidades poderia ser disponibilizado em apenas um ecrã;
- A informação da localização dos motoristas podia ser complementada com a lista de pacotes que o mesmo se encontra a entregar e/ou recolher;
- Os marcadores utilizados para a identificação dos motoristas no mapa podiam utilizar as cores do modo de entrega em que o motorista se encontra;
- A aplicação devia relacionar a lista de motoristas com os marcadores no mapa.

4.6 Conclusões

A disponibilização deste conjunto de funcionalidades no HMT permitiu levar até a um grupo de utilizadores a perspetiva de uma solução que lhes poderá disponibilizar o acesso eficiente à informação da localização de motoristas.

A realização do inquérito de satisfação permitiu compreender a opinião dos utilizadores, no que toca à facilidade com que utilizam a aplicação. O facto de os utilizadores considerarem a

aplicação amigável, é um passo importante para que sintam entusiasmo e que tirem partido da informação prestada pela mesma.

No que toca à utilidade e à frequência com que os utilizadores pretendem utilizar que as novas funcionalidades, em que mais de metade dos inquiridos as consideraram úteis e assumem pretender utiliza-las pelo menos uma vez por dia, permite-nos concluir que este protótipo está no caminho certo para a melhoria de desempenho dos colaboradores.

Por fim, no final do inquérito foi dado espaço aos utilizadores para deixarem sugestões de melhoria da aplicação. Esta área e a atenção dada à opinião dos utilizadores permite-nos levar as próximas versões do protótipo mais próximas das necessidades dos utilizadores.

De um modo geral, pode concluir-se que a disponibilização da informação dos motoristas aos operadores dos armazéns será um passo importante na melhoria do seu desempenho diário. É ainda possível compreender que a criação de um novo módulo no HMT para a disponibilização destas funcionalidades é uma boa opção para a facilidade de acesso à informação.

5 Conclusão

Neste capítulo é feita uma reflexão acerca do trabalho realizado com uma abordagem genérica das conclusões (secção 5.1) e a identificação de perspetivas da evolução e utilização do estudo realizado (secção 5.2).

5.1 Resumo do trabalho realizado

Depois de realizado este projeto, com base nos resultados obtidos e nos comentários por parte dos restantes colaboradores da empresa em que o mesmo foi realizado, o resultado final é classificado como sucesso.

A realização do estudo na primeira parte do projeto permitiu que fosse identificada a melhor solução para o armazenamento de informação geográfica dos motoristas da empresa. Este estudo é um importante apoio na tomada de decisão da tecnologia a adotar para o desenvolvimento de soluções de localização de motoristas e, eventualmente, pacotes que se encontrem a ser entregues. Contudo considera-se que, idealmente, outras soluções de armazenamento poderiam ter sido consideradas no processo de análise de desempenho.

Com este estudo, conclui-se que as diferentes Bases de Dados analisadas apresentaram resultados distintos durante a escrita e a leitura de informação. Foi, assim, possível concluir que o Elasticsearch representa uma melhor solução na leitura de informação enquanto que o PostgreSQL surgiu com melhores resultados durante a escrita de dados. No que toca à escrita de informação, ambas as soluções aumentaram consideravelmente o desempenho quando o registo dos dados foi feito no modo BULK.

Relativamente ao novo módulo acrescentado ao HMT, esta é uma secção que permite aos utilizadores da aplicação ter um acesso rápido à localização dos motoristas a operar nos seus armazéns. Com o inquérito final realizado foi possível verificar, de um modo geral, a satisfação dos mesmos. O mesmo inquérito permitiu identificar pontos de melhoria a serem realizados na aplicação no sentido de tornar o produto mais útil e funcional, nomeadamente na disponibilização de novas funcionalidades, realização de um mapeamento visual entre a lista de

motoristas e os marcadores no mapa, aumento do detalhe relativamente aos motoristas, entre outras.

De um modo geral, a realização deste trabalho foi um passo importante no desenvolvimento de uma aplicação de tracking de motoristas e encomendas, na medida em que o estudo realizado e a prova de conceito serão tidos em conta na escolha de tecnologias e implementação de uma versão final das funcionalidades.

5.2 Trabalho futuro

Na sequência do desenvolvimento deste projeto pretende-se avançar para o desenvolvimento de uma solução definitiva de *tracking* de motoristas. O desenvolvimento desta solução partirá da prova de conceito realizada durante o projeto descrito no documento, sendo aplicadas as melhorias sugeridas pelos utilizadores nas respostas ao inquérito.

Prevê-se que a prova de conceito evolua para uma funcionalidade numa versão mais madura e que seja disponibilizada a um número mais alargado de utilizadores.

6 Referências

Africa Internet Group, 2016. *About us | AfricaInternetGroup*. [Online]

Available at: <http://africainternetgroup.com/about-us>

Anon., 2015. *PostGIS in Action*. Second ed. Shelter Island: Manning Publications.

Boral, H. & DeWitt, D., s.d. *A methodology for Database System Performance Evaluation*, s.l.: s.n.

Gudivada, V., Rao, D. & Raghavan, V., 2014. *Renaissance in Data Management Systems: SQL, NoSQL, and NewSQL*. Huntington; Lafayette, s.n.

Nance, C., Iype, R., Lossner, T. & Harmon, G., 2013. *NOSQLVS RDBMS - WHY THERE IS ROOM FOR BOTH*. Savannah, s.n.

PANT, N., 2015. *Performane comparison of spatial indexing structures for different query types*, s.l.: s.n.

Power, R., 2009. *Testing geospatial database implementations for water data*. Cairns, s.n.

Tauro, C., A. & S., s.d. *Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases*.

A1. Inquérito de satisfação

HMT tracking satisfaction survey	
1. How user-friendly is our software's interface?	
Extremely user-friendly Very user-friendly Moderately user-friendly Slightly user-friendly Not at all user-friendly	
2. Are you satisfied with the new features provided by the application?	
Yes No	
3. Does the information in the new screen are helpful for you daily job?	
Yes No Not applicable	
4. Do you think that these new features might be helpful to improve the performance of you hub?	
Yes No Not applicable	
5. As a hub manager, how often do you pretend to use this features to monitor you drivers?	
Once a day Twice a day More than three times a day Never Not applicable	
6. Use this area to leave some suggestions	