



Universidade de Aveiro  
2018

Departamento de Eletrónica,  
Telecomunicações e Informática

Tiago Filipe Maio  
Teixeira

## Representação e análise de informação contextual georreferenciada em sistemas de monitorização de desempenho em redes

Relatório de dissertação/projeto/estágio apresentado à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor José Manuel Matos Moreira, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro



o júri

presidente

**Prof. Doutor Joaquim João Estrela Ribeiro Silvestre Madeira**  
professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da  
Universidade de Aveiro

**Prof. Doutora Mariana Curado Malta**  
professora adjunta do Instituto Politécnico do Porto

**Prof. Doutor José Manuel Matos Moreira**  
professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da  
Universidade de Aveiro



palavras-chave

Análise de dados georreferenciados, informação contextual, indicadores-chave de desempenho em redes de telecomunicações

resumo

O negócio de telecomunicações é uma área que está em constante evolução e, com o crescimento e o surgimento de novas tecnologias, as operadoras precisam de encontrar novas soluções que satisfaçam os seus clientes. Este trabalho consiste num estudo exploratório realizado em parceria com a Nokia Solutions and Networks, cujo objetivo principal é cruzar informação da ferramenta Nokia Performance Manager com informações externas que possam acrescentar valor, por exemplo, dados meteorológicos, fluxo de tráfego e eventos sociais. Para tal, foi desenvolvida uma solução para modelar e integrar este tipo de informação contextual no sistema de monitorização do desempenho de redes atual, dando particular importância às dimensões geográfica e temporal, e foram implementados processos de recolha periódica e extração de informação contextual a partir de serviços disponíveis online. Uma vez que os dados são recolhidos de forma periódica e que o número de elementos a monitorar numa rede de telecomunicações é elevado, foi feita uma avaliação de desempenho e foram testadas diversas soluções para melhorar os tempos de resposta do sistema. O resultado obtido foi um sistema capaz de representar a informação da rede e informação contextual, em que os tempos de resposta são compatíveis com a realização de tarefas de análise de dados interativas, utilizando para tal um mecanismo de cache. A informação pode ser filtrada e visualizada com base nas suas dimensões contextual, geográfica e temporal.



## Keywords

Geospatial data analysis, contextual information, key performance indicators in telecommunication networks

## Abstract

The telecommunications business is an area that is constantly evolving and with the growth and appearance of new technologies, operators must find new ways to exploit and find solutions that will please their customers. This report deals with an exploratory study carried out in partnership with Nokia Solutions and Networks, whose main objective is to cross-reference information from its existing NPM tool with external information that may be of added value to the business, for example: meteorological data, traffic flow and social events. To do that, a solution to model and integrate this type of contextual information into the current network performance monitoring system was developed, giving importance to the geographic and temporal dimensions, and the processes for periodic collection and extraction of contextual information from online services were implemented. Since data are collected periodically and the number of elements to be monitored in a telecommunications network is high, a performance assessment has been made and various solutions have been tested to improve system response times. The result was a system capable of representing the network information and contextual information, in which the response times are fast enough to enable interactive data analysis, by using a caching mechanism. Information can be filtered and viewed based on its contextual, geographical and temporal dimensions.





# Índice

Índice.....	i
Lista de figuras.....	v
Lista de tabelas.....	vii
Lista de acrónimos.....	ix
1 Introdução.....	1
1.1 Contexto.....	1
1.2 Motivação.....	2
1.3 Objetivo.....	2
1.4 Estrutura do relatório.....	4
2 Sistemas de apoio à decisão espaciais e serviços de informação online.....	5
2.1 Data warehouses espaciais.....	5
2.1.1 Contextualização.....	5
2.1.2 Sistemas de processamento analítico de dados espaciais.....	6
2.1.3 Propostas de sistemas de processamento analítico de dados espaciais.....	8
2.2 Servidores de mapas.....	10
2.3 Fontes externas.....	12
2.4 Síntese.....	13
3 Requisitos do sistema.....	15
3.1 Sistema atual.....	15
3.2 Requisitos.....	16
3.2.1 Requisitos funcionais.....	16
3.2.2 Requisitos não funcionais.....	18

3.3	Funcionalidades dos módulos.....	18
3.3.1	Base de dados.....	19
3.3.2	Cliente dos serviços web.....	19
3.3.3	Camada intermédia.....	20
3.3.4	Protótipo de demonstração.....	21
3.4	Atores e casos de uso.....	21
3.5	Síntese.....	22
4	Arquitetura e implementação.....	23
4.1	Arquitetura do sistema.....	23
4.2	Base de dados.....	24
4.2.1	Topologia da rede e indicadores de performance (KPIs).....	25
4.2.2	APIs externas.....	26
4.3	Simulador de dados.....	28
4.3.1	Topologia de rede.....	28
4.3.2	Valores dos indicadores de desempenho.....	29
4.4	APIs externas.....	30
4.4.1	Meteorologia.....	31
4.4.2	Trânsito.....	31
4.4.3	Eventos.....	32
4.5	Cliente das APIs e agendamento de processos.....	34
4.5.1	Cliente das APIs externas.....	34
4.5.2	Conexão à base de dados.....	35
4.5.3	Agendamento de processos.....	35
4.5.4	Comunicação entre módulos.....	36
4.6	Geoserver e GeoWebCache.....	37
4.6.1	Geoserver.....	38
4.6.2	GeoWebCache.....	42
4.7	Interface do cliente.....	45

4.8 Síntese.....	48
5 Testes e resultados.....	49
5.1 Funcionalidade.....	49
5.1.1 Ambiente de testes e dados utilizados .....	49
5.1.2 Histórias de uso .....	50
5.2 Persistência .....	51
5.3 Desempenho .....	54
5.4 Discussão.....	56
6 Conclusão e trabalho futuro .....	57
7 Referências .....	61
8 Apêndices.....	65



# Lista de figuras

Figura 1- Análise do sucedido (esquerda) e análise do futuro (direita) .....	3
Figura 2 - Posição do GIS e SOLAP no que diz respeito a três eixos de requisitos para o suporte à decisão espacial., retirado de [5] .....	7
Figura 3 - Os três tipos de dimensões espaciais, retirado de [10] .....	8
Figura 4 - Hierarquia de rede de telecomunicações .....	16
Figura 5 - Representação da origem dos dados .....	17
Figura 6 - Arquitetura conceptual do projeto .....	19
Figura 7 - Funcionamento da camada intermédia .....	20
Figura 9 - Casos de uso do projeto .....	22
Figura 10 - Arquitetura do sistema.....	23
Figura 11 - Esquema da base de dados dos elementos de rede .....	26
Figura 13 - Esquema de base de dados das fontes externas .....	27
Figura 14 - Fluxo de processamento de script de criação de topologia de rede .....	29
Figura 15 - Fluxo do processamento de script de simulador de dados.....	30
Figura 17 – Resultado do pedido da API Darksky .....	31
Figura 18 - Resultado do pedido da API Here .....	32
Figura 19 - Resultado do pedido da API Eventful .....	33
Figura 20 - Diagrama de classes do cliente das APIs.....	35
Figura 21 - Diagrama de classes da conexão à base de dados.....	35
Figura 22 - Diagrama de classes do agendamento de processos .....	36
Figura 23 - Diagrama da comunicação entre módulos.....	37
Figura 24 - Interface do Geoserver .....	38
Figura 25 - Configuração da conexão à base de dados no Geoserver .....	39
Figura 26 - Exemplo de estilo, retirada de [49].....	40
Figura 27 - Layer do trânsito fluído do dia 2 de maio de 2018 às 16 horas .....	41
Figura 28 - Ícones da meteorologia.....	41

Figura 29 - Exemplo de um layer KPI .....	42
Figura 30 - Configuração do cache de um layer .....	43
Figura 31 - Configuração da base de dados do GeoWebCache .....	44
Figura 32 - Pedido POST para pré-cálculo de layer.....	45
Figura 33 - Pedido WMS no Leaflet, retirado de [50] .....	46
Figura 34 - Filtros da interface final.....	46
Figura 35 – Recorte da apresentação final da interface.....	47
Figura 36 - Seleção de elemento .....	47
Figura 37 - Cobertura geográfica dos diferentes contextos.....	50
Figura 38 - Armazenamento diário na base de dados .....	53

## Lista de tabelas

Tabela 1 – Plataforma e representação dos projetos SOLAP .....	9
Tabela 2 - Funcionalidades dos projetos de SOLAP .....	10
Tabela 3 - Comparação entre Geoserver e MapServer .....	11
Tabela 4 - Lista de APIs de meteorologia .....	12
Tabela 5 - Lista de APIs de trânsito .....	12
Tabela 6 - Lista de APIs de eventos .....	13
Tabela 7 - Características dos diferentes níveis de zoom .....	52
Tabela 8 - Espaço ocupado por diferentes formatos de tiles .....	52
Tabela 9 - Espaço de armazenamento de difentes intervalos de tempo .....	54





# Lista de acrónimos

<b>NSN</b>	Nokia Solutions and Network
<b>NPM</b>	Nokia Performance Manager
<b>KPI</b>	Key Performance Indicator
<b>CSV</b>	Comma Separated Values
<b>API</b>	Application Programming Interface
<b>GIS</b>	Geographic Information System
<b>OLAP</b>	Online Analytical Processing
<b>SOLAP</b>	Spatial Online Analytical Processing
<b>OGC</b>	Open Geospatial Consortium
<b>WMS</b>	Web Map Service
<b>WFS</b>	Web Feature Service
<b>WCS</b>	Web Coverage Service
<b>REST</b>	Representational State Transfer
<b>SMS</b>	Short Message Service
<b>BSC</b>	Base Station Controller
<b>RNC</b>	Radio Network Controller
<b>PLMN</b>	Public Land Mobile Network
<b>PNG</b>	Portable Network Graphics
<b>JPEG</b>	Joint Photographic Experts Group

<b>GIF</b>	Graphics Interchange Format
<b>JDBC</b>	Java Database Connectivity
<b>SQL</b>	Structured Query Language
<b>SLD</b>	Styled Layer Descriptor
<b>XML</b>	Extensible Markup Language
<b>JSON</b>	JavaScript Object Notation
<b>HTTP</b>	Hypertext Transfer Protocol

# 1 Introdução

## 1.1 Contexto

A rápida evolução da tecnologia faz com que seja cada vez mais importante para as operadoras de telecomunicações terem capacidade para responder às necessidades dos seus utilizadores, que cada vez se tornam mais exigentes em relação à qualidade dos serviços contratados. Assim sendo, as operadoras de telecomunicações necessitam de ter formas abrangentes de analisar e perceber as alterações de utilização da sua rede, por forma a poderem agir atempadamente e do modo mais eficaz possível na prevenção e deteção de problemas.

A Nokia Solutions and Networks (NSN), sendo líder de mercado em equipamentos, soluções e serviços para redes de banda larga móvel, tem a necessidade de garantir uma qualidade de excelência nos seus produtos por forma a servir alguns dos maiores operadores de telecomunicações a nível mundial. A empresa opera em mais de 150 países e conta com mais de 60.000 colaboradores em todo o mundo. Em Portugal, nomeadamente nos centros tecnológicos localizados em Lisboa e Aveiro, colaboram cerca de 2.000 pessoas [1]. Parte deste sucesso da empresa deve-se à ferramenta Nokia Performance Manager (NPM) que desempenha um papel fundamental no portefólio que a Nokia oferece.

O NPM é responsável por coletar, processar e apresentar de forma imediata e unificada uma enorme quantidade de informação proveniente de redes de pequena e larga escala. Desta forma o NPM tem um papel essencial na prevenção e deteção prévia de problemas na rede, possibilitando assim uma tomada de decisão mais consciente e eficaz aos operadores, que por sua vez lhes permite garantir uma melhor qualidade de serviço aos seus clientes finais. Através dela é possível ter acesso a um conjunto de funcionalidades, tais como: visualização de relatórios em tabelas e gráficos; relatórios que incluem os indicadores-chave de desempenho (KPI); exportação para Excel e CSV (Comma Separated Values) com vistas em tabelas e gráficos; relatórios de navegação; agendamento de relatórios; filtragem e pesquisa avançada, entre outras. De sublinhar que, aos dados provenientes

de redes de banda larga móvel está associada uma componente geográfica que corresponde à posição de cada antena de telecomunicações [2].

## **1.2 Motivação**

O projeto descrito no presente relatório de estágio foi proposto pela Nokia Solutions and Network e centra-se na forma como as operadoras analisam e tomam decisões sobre a informação que têm à sua disposição. Os sistemas de análise de rede atuais são essenciais para que as operadoras percebam como se encontrava o estado da rede num determinado momento. No entanto, estes sistemas apenas têm à sua disposição a informação da rede de telecomunicações, percebendo assim quando e como se encontra o estado de cada elemento da rede, mas não sendo possível perceber se um valor negativo teve origem em contextos externos à rede ou não. Por esta razão, a premissa inicial para este estudo é ajudar o analista a perceber se o estado dos elementos da rede foi influenciado por fatores externos e, conseqüentemente, percebendo a origem de determinado problema, tentar alcançar uma solução que resolva ou contorne o problema levantado.

## **1.3 Objetivo**

Tendo como objetivo desenvolver e criar formas de extrair informação mais refinada e complexa, pretende-se analisar e cruzar os dados da rede com outras fontes de informação disponíveis e fazer a representação georreferenciada dessa informação. Assim, utilizando esta informação da rede, cruzando os dados do passado com previsões futuras e dando a possibilidade de integrar outras fontes de informação externas à rede, pode ser dada aos operadores de telecomunicações uma forma mais abrangente e eficaz de visualizar, navegar e analisar os dados de operação da rede.

Pretende-se que, utilizando os dados já existentes na ferramenta NPM, se desenvolva um estudo para tornar possível cruzar esta informação com fontes de informação externa (por exemplo: meteorologia, fluxo de trânsito, eventos a decorrer) para que os especialistas em gestão de redes possam perceber de que forma estes novos dados podem ser uma mais valia para a análise da rede, por exemplo para detetar ou antecipar anomalias na rede devido a condições meteorológicas excepcionais, fluxo de trânsito pouco fluído ou eventos de grande dimensão, como ilustrado na Figura 1. Perante a ocorrência de uma falha (à esquerda na figura), é tido em análise os possíveis contextos que possam ter tido influência para que esta ocorresse e à direita são analisadas as condições contextuais e comparadas com dados históricos de forma a prever falhas que possam vir a suceder. Isto é, criar um sistema de apoio à decisão com esta informação.

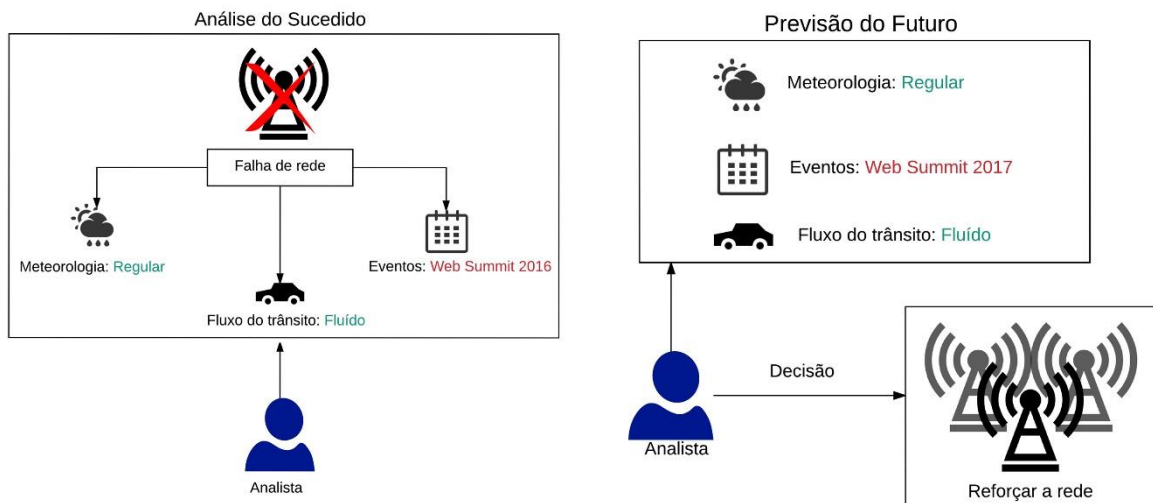


Figura 1- Análise do sucedido (esquerda) e análise do futuro (direita)

Consequentemente, o trabalho proposto pode ser dividido em 4 problemas distintos:

- **Como obter informação?**

Pretende-se perceber como é que a informação deve ser obtida, tratada, armazenada e utilizada para desenvolver e criar formas de extrair informação mais refinada e complexa. Em seguida, explorar as APIs (Application Programming Interface) existentes para obter informação externa (eventos, trânsito e meteorologia) que possa complementar os dados já existentes e permitir assim uma análise mais detalhada da rede.

- **Como tratar e armazenar a informação?**

Uma vez reunidas as fontes de informação necessárias e tendo uma considerável compreensão dos dados será definida a forma ideal para os transformar e armazenar. Para isto é necessário ter em conta a correlação dos dados, a forma como serão disponibilizados e apresentados e os casos de uso dos mesmos.

- **Como disponibilizar a informação?**

Nesta fase o trabalho deverá focar-se na forma como esta informação é disponibilizada para ser apresentada, e na forma mais eficiente de o fazer.

- **Como apresentar a informação?**

Nesta etapa deve ser encontrada uma forma de demonstrar os resultados obtidos como prova de conceito. Isto é, para efeitos de demonstração da solução deve ser criada uma interface em que seja apresentada a informação geográfica recolhida.

## 1.4 Estrutura do relatório

O trabalho desenvolvido será apresentado ao longo de seis capítulos de forma detalhada. No primeiro foi apresentada uma contextualização do trabalho proposto, nomeadamente a empresa NSN e a sua ferramenta NPM, e foi apresentada a motivação para o presente estudo e os seus objetivos. No segundo capítulo apresentam-se conceitos e trabalhos anteriores sobre armazenamento de dados espaciais e processamento analítico de dados georreferenciados, assim como, servidores de mapas e as fontes externas existentes que podem melhor atender às necessidades deste trabalho. Nos dois capítulos subsequentes são apresentados os principais requisitos do sistema e a sua implementação, respetivamente. Aquando da abordagem dos requisitos são explorados os cenários que servem de premissa para o presente estudo, assim como, o que cada módulo do sistema deve ser capaz de fazer. No quinto capítulo são expostos os resultados obtidos no desenvolvimento do sistema, assim como uma breve discussão sobre algumas implementações do mesmo. Por fim, no sexto capítulo, são redigidas as conclusões e são propostas ideias para trabalho futuro que podem vir a ser significativas no desenvolvimento deste protótipo.

## 2 Sistemas de apoio à decisão espaciais e serviços de informação *online*

Esta secção apresenta o estado da arte sobre representação e análise de informação contextual georreferenciada em sistemas de monitorização de desempenho em redes. No tópico 2.1 é apresentada uma introdução sobre *data warehouses* e Geographic Information System (GIS), que são basilares para a compreensão e domínio do que é apresentado em seguida, *data warehouses* espaciais, onde é possível compreender em que é que o *Spatial Online Analytical Processing* (SOLAP) consiste e por fim são apresentados alguns protótipos criados para este tema. Na secção 2.2 são expostos os protocolos utilizados para disponibilizar informação georreferenciada e o que é um servidor de mapas, sendo apresentadas algumas das soluções existentes no mercado atualmente. Por fim, no tópico 2.3, são apresentadas e descritas algumas das APIs que podem vir a ser úteis para o desenvolvimento do trabalho proposto.

### 2.1 Data warehouses espaciais

#### 2.1.1 Contextualização

Um *data warehouse* (DW) é um conjunto de tecnologias que permitem uma melhor e mais fácil tomada de decisão. Esta definição foi apresentada por S. Chaudhuri e U. Dayal [3] e, segundo os autores, um DW é direcionado para ser utilizado por parte de um trabalhador executivo, gestor ou analista. É normalmente composto por um agregado de informação que é mantido à parte das bases de dados operacionais de uma organização. Por ser direcionado para trabalhadores com cargos de análise, a informação é organizada temporalmente, sumariada e consolidada. Por este motivo, os dados são guardados por longos períodos de tempo e o tamanho de um *data warehouse* é sempre projetado para ter centenas de gigabytes ou até alguns terabytes. Ao contrário de uma base de dados relacional, onde os dados guardados se focam nos detalhes mais atualizados e individuais de cada registo, um DW suporta processamento analítico online (OnLine Analytical Processing - OLAP) que

foi descrito por J. Han, M. Kamber e J. Pei [4] como um conjunto de técnicas de análise com funcionalidades de consolidação, agregação e sobretudo com capacidade de analisar a informação de diferentes pontos de vista. Além disto, o OLAP também suporta ferramentas de análise multidimensional e de tomada de decisão.

No entanto, os dados provenientes da ferramenta NPM têm uma componente geográfica, que não pode ser ignorada e deve ser tratada dessa forma, como uma dimensão espacial. S. Rivest, Y. Bédard e P. Marchand [5] demonstraram que é possível utilizar sistemas OLAP (não espaciais) para suportar análise de dados espaço-temporal com sucesso. Porém, apesar de, segundo os autores, o OLAP oferecer bom suporte para o uso simultâneo de dimensões temporais e espaciais para o processo de análise, estes acrescentam que as dimensões espaciais são tratadas como outra dimensão qualquer, sem considerar a sua componente geográfica. Sendo esta uma das maiores limitações de um sistema OLAP neste campo, a par da falta de possibilidade de ter uma visualização espacial dos dados, é a impossibilidade de fazer uma análise da informação com base na exploração de um mapa.

Por outro lado, os sistemas de informação geográfica (Geographic Information System - GIS) são ferramentas baseadas em base de dados relacionais e têm como principal objetivo a visualização e exploração de dados espaciais [7]. Segundo o estudo conduzido por MacEachren [8], os mapas interativos e conteúdos gráficos são as principais técnicas e as mais eficientes para a análise de informação espaço-temporal, podendo assim revelar padrões espaciais e estimular o processo de pensamento do analista. Contudo, a maioria das aplicações GIS foram desenhadas usando uma abordagem baseada em transações e em bases de dados relacionais. Desta forma, a maneira como estas ferramentas estão otimizadas dificulta a tomada de decisão [7].

Consequentemente, para preencher esta lacuna tanto nos sistemas DW e OLAP como nos sistemas GIS, surgiu o conceito de *data warehouses* espaciais e, como muitas vezes chamado na literatura, SOLAP.

### **2.1.2 Sistemas de processamento analítico de dados espaciais**

O acrónimo SOLAP pode ser dividido em “Spatial” “OLAP”, isto é: um sistema OLAP especializado para a representação de dados espaciais. Foi idealizado para preencher as lacunas das ferramentas OLAP, que não suportam a visualização de dados geográficos e dos sistemas GIS, que não estão direcionados para a análise e tomada de decisões de dados espaço-temporal. O primeiro conceito de SOLAP apresentado por Bédard [9] em 1997, é descrito como um sistema que suporta a análise espaço-temporal e a exploração dos dados, agregando assim todos os níveis disponíveis e seguindo uma aproximação multidimensional tanto em mapas como também em tabelas e diagramas. Por outro lado, S. Rivest, Y. Bédard e P. Marchand [5] defendem que o SOLAP deve ser um sistema



de utilização elementar e de rápida e fácil aprendizagem, contrariamente aos sistemas GIS, como é sugerido na Figura 2.

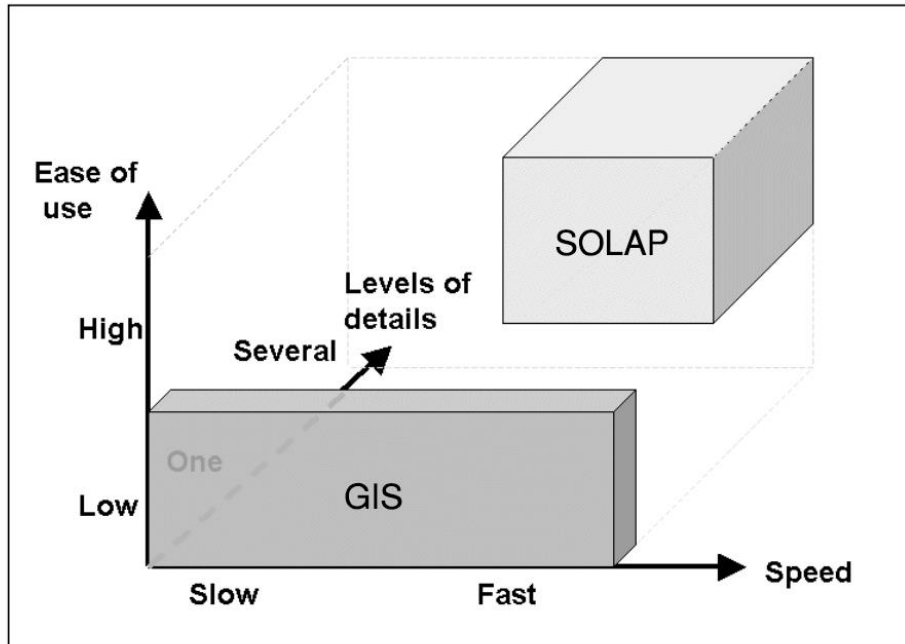


Figura 2 - Posição do GIS e SOLAP no que diz respeito a três eixos de requisitos para o suporte à decisão espacial., retirado de [5]

O SOLAP também é composto por uma base de dados estruturada, um servidor e um cliente, à imagem do OLAP. No entanto a sua base de dados deve suportar dados espaciais, assim como o servidor SOLAP deve ser compatível com a utilização deste tipo de dados [11].

Para modelar bases de dados orientadas para SOLAP é necessário que estas suportem dados espaciais e, para este fim, Jiawei Han, Nebojsa Stefanovic e Krzysztof Koperski [10], afirmam que há três formas distintas para modelar a dimensão espacial num cubo:

- dimensão não-espacial (*nonspatial*): contém apenas dados não espaciais (ex.: “temperatura”);
- dimensão espacial para espacial (*spatial to spatial*): é aquela em que o nível primitivo é espacial assim como toda a sua informação;
- dimensão espacial para não-espacial (*spatial to nonspatial*): dados em que o nível primitivo é espacial, mas em que a generalização a partir de certo nível se torna não espacial. Ex: um

estado no mapa dos Estados Unidos da América é representado de forma espacial, no entanto cada um destes pode ser traduzido num dado não-espacial.

Estas dimensões foram categorizadas por S. Rivest [11] como não-geométricas, geométricas e mistas, e é possível perceber melhor a respetiva estrutura na Figura 3.

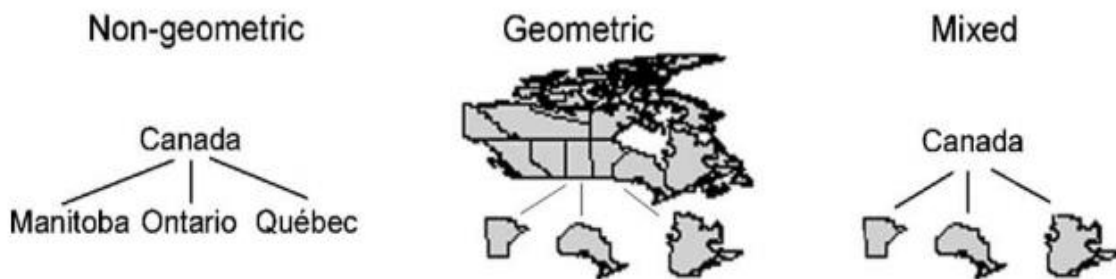


Figura 3 - Os três tipos de dimensões espaciais, retirado de [11]

### 2.1.3 Propostas de sistemas de processamento analítico de dados espaciais

Ao longo dos anos, depois do conceito de *data warehouses* com dados geográficos surgir, foram apresentadas várias propostas e protótipos na literatura. Neste tópico vamos explorar as características e as funcionalidades que os autores foram melhorando ao longo dos anos neste tipo de propostas. Para esta análise foram escolhidos projetos dentro de um intervalo de 20 anos, ou seja, desde que este tema começou a ganhar algum destaque até aos dias de hoje.

Na Tabela 1 estão listados os projetos escolhidos para esta análise e, indo ao encontro da informação nela apresentada, é possível verificar que uma das características que definem os primeiros projetos é serem desenvolvidos para ser executados em *desktop*. No entanto, esta tendência foi-se alterando ao longo dos anos e a plataforma que tem vindo a ser adotada é a *web*. Porém, em relação à utilização de mapas em 2D ou 3D, não se pode considerar que seja uma tendência a utilização de mapas em 3D, mas sim mais uma opção a considerar aquando do desenvolvimento de um sistema SOLAP.

Tabela 1 – Plataforma e representação dos projetos SOLAP

<b>Autor (data)</b>	<b>Plataforma</b>	<b>Mapa</b>
Han (1997) [12]	<i>Desktop</i>	2D
Shekhar (2000) [13]	<i>Desktop</i>	2D
Stolte (2003) [14]	<i>Desktop</i>	2D
Fidalgo (2004) [20]	<i>Web</i>	2D
Colonese (2005) [15]	<i>Desktop</i>	2D
Parmanto (2005) [16]	<i>Desktop</i>	2D
Compieta (2007) [17]	<i>Desktop</i>	3D
Bimonte (2007) [18]	<i>Web</i>	2D
Silva (2008) [19]	<i>Web</i>	2D
Ferraz (2010) [21]	<i>Web</i>	3D
Di Martino (2011) [22]	<i>Web</i>	3D
Derbal (2016) [23]	<i>Web</i>	2D
Gur (2017) [24]	<i>Web</i>	2D

Na Tabela 2 estão listados estes mesmos projetos, mas agora com as funcionalidades que foram sendo adicionadas a estes sistemas ao longo dos anos:

- valores KPI: possibilidade de saber os valores dos KPI em estudo, funcionalidade que seria indispensável a qualquer sistema SOLAP;
- regiões diferenciadas com cores: cada região do mapa é pintada de uma cor diferente consoante o seu valor de KPI ou uma característica específica;
- níveis de *zoom*: possibilidade de aumentar ou diminuir o detalhe apresentado no mapa;
- gráficos: possibilidade de ver gráficos de barras ou circulares com estatísticas dos valores de KPI em estudo;
- explorar o mapa: possibilidade de o explorar com o cursor.

Tabela 2 - Funcionalidades dos projetos de SOLAP

Autor (data)	Valores KPI	Regiões diferenciadas c/ cores	Níveis de zoom	Gráficos	Explorar o mapa
Han (1997) [12]	✓				
Shekhar (2000) [13]	✓	✓			
Stolte (2003) [14]	✓	✓	✓		
Fidalgo (2004) [20]	✓				
Colonese (2005) [15]	✓				
Parmanto (2005) [16]	✓	✓		✓	
Compieta (2007) [17]	✓		✓		✓
Bimonte (2007) [18]	✓	✓		✓	
Silva (2008) [19]	✓		✓		✓
Ferraz (2010) [21]	✓	✓	✓	✓	✓
Di Martino (2011) [22]	✓		✓	✓	✓
Derbal (2016) [23]	✓	✓	✓	✓	✓
Gur (2017) [24]	✓		✓		✓

Desta forma, podemos concluir que ao longo dos anos houve funcionalidades que se foram tornando indispensáveis no desenvolvimento destes sistemas, mas, por outro lado, há outras que são apenas opcionais.

## 2.2 Servidores de mapas

A representação de informação georreferenciada *online* não é nova, desde o início dos anos 90 que esta é apresentada na *web*, no entanto, de forma muito simples, apenas produzindo mapas com imagens estáticas. Entretanto com o aparecimento do JavaScript estas representações foram-se tornando cada vez mais interativas [25].

O rápido desenvolvimento desta tecnologia e o aparecimento de inúmeras aplicações para representação de mapas *web* fez com que a necessidade de criar protocolos e padrões de representação aumentasse. Assim, o Open Geospatial Consortium (OGC) publicou em 1999 o seu primeiro manual de implementações [26], que depois de revisto se viria a tornar, em 2000, o Web Map Server Interface Implementation Specification (versão 1.0) que ao longo do tempo e de novas

revisões definiu um padrão para a forma como os serviços de mapas são implementados [27]. A OGC definiu protocolos como:

- Web Map Service (WMS) que transforma informação georreferenciada em ficheiros de imagem, onde cada uma representa uma parte do mapa (*tile*), fornecendo mapas/camadas estáticos ao cliente (*layer*) [27].
- Web Feature Service (WFS) que, ao contrário de WMS, fornece ao cliente as características da informação georreferenciada solicitada num ficheiro. No entanto renderizar os dados fornecidos ao cliente (dependendo do tamanho do ficheiro) leva consideravelmente mais tempo que receber as imagens já renderizadas do WMS, em contrapartida é possível os clientes alterarem algum tipo de informação solicitada [28].
- Web Coverage Service (WCS) que suporta a recuperação eletrónica de dados geoespaciais (dados georreferenciados que representam fenómenos que variam no espaço). O WCS devolve os dados georreferenciados originais juntamente com os seus atributos associados, proporcionando assim a oportunidade para exploração e interpretação de dados [29].

Os servidores de mapas tornam possível aceder a informação georreferenciada presente numa base de dados e disponibilizá-la utilizando os protocolos da OGC referidos acima. Para esta solução há duas ferramentas a considerar: Geoserver e MapServer, tendo como base o estudo desenvolvido por Brock Anderson [30] e Marc Jansen [31], foi concluído que os dois servidores de mapas têm funcionalidade muito idênticas. Na Tabela 3 são apresentadas algumas das funcionalidades destas duas aplicações.

*Tabela 3 - Comparação entre Geoserver e MapServer*

<b>Software</b>	Geoserver [32]	MapServer [33]
<b>Linguagem</b>	Java	C/C++
<b>Padrão WMS</b>	✓	✓
<b>Padrão WFS</b>	✓	✓
<b>Protocolo WCS</b>	✓	✓
<b>Bases de Dados Compatíveis</b>	PostGIS, MySQL, Oracle, SQL Server, H2, ArcSDE, DB2	PostGIS, MySQL, Oracle, ESRI shapfiles, SpatiaLite, ESRI ArcSDE
<b>Custo</b>	Grátis	Grátis
<b>Última Versão (Data)</b>	2.13.1 (25-04-2018)	7.0.7 (15-11-2017)

## 2.3 Fontes externas

As fontes externas são os dados que vão ser cruzados com os da ferramenta NPM com o intuito de tirar conclusões. Os temas propostos para análise foram meteorologia, trânsito e eventos sociais, sendo estes fatores que podem condicionar o desempenho da rede de telecomunicações. Para tal, serão usadas APIs Representational State Transfer (REST) para obter a informação de cada um dos serviços pretendidos. Nas tabelas seguintes estão listadas as principais APIs para cada um dos temas, seguidas de alguns dos seus detalhes:

*Tabela 4 - Lista de APIs de meteorologia*

<b>Meteorologia</b>		
API	N.º de pedidos	Preço
Aeris Weather [34]	5k   50k   100k   250k dia	25\$   85\$   150\$   350\$ mês
Open Weather Map [35]	60   600   3k   30k   200k minuto	Grátis   40\$   180\$   470\$   2000\$ mês
Dark Sky [36]	1k dia	Grátis + 0.0001\$ p/ pedido
HERE Weather API [37]	150k+ mês	Não é apresentado
World Weather Online [38]	500   5k   15k   50k   100k   500k   1M dia	9\$   15\$   20\$   30\$   42\$   60\$   77\$ mês

No contexto meteorológico todos os serviços analisados fornecem o mesmo tipo de informação no que diz respeito ao estado atual do tempo numa determinada localização, como por exemplo, temperatura atual, velocidade do vento, humidade, etc., sendo comum a quase todos, o preço de utilização de cada API aumentar consoante o número de pedidos que são efetuados. Todas as versões pagas dão acesso ao histórico de temperaturas seja da semana passada ou de todo o mês, mas este tipo de funcionalidade não traz valor ao projeto proposto.

*Tabela 5 - Lista de APIs de trânsito*

<b>Trânsito</b>		
API	N.º de pedidos	Preço
Here Transit API [37]	150k+ mês	Não é apresentado
Google Maps API* [39]	25k dia	Grátis
TransLink Open API** [40]	Ilimitado	Grátis
WalkScore** [41]	5000  ilimitado	Grátis   100\$

\*API em JavaScript

\*\*não disponível em Portugal

Em relação a serviços de trânsito, das APIs analisadas, apenas a Here Transit API e a Google Maps API têm disponíveis informação no território português. No entanto, a API da Google funciona como uma camada (*layer*) a adicionar por cima do mapa, pintando as ruas de verde, amarelo e vermelho, isto é, trânsito fluído, pouco fluído e congestionado, respetivamente. Este serviço obriga a que o Google Maps seja o serviço de mapas escolhido para que esta representação seja possível. Por outro lado, no caso da API da Here, os dados devolvidos são pedidos usando a respetiva API REST, definida pela seguinte informação para cada troço de estrada: a qualidade da estrada, velocidade praticada, velocidade permitida, código da estrada, pontos que definem a estrada e o nome da estrada.

*Tabela 6 - Lista de APIs de eventos*

Eventos		
API	N.º de pedidos	Preço
Eventful [42]	Ilimitado	Grátis
Bandsintown*** [43]	Ilimitado	Grátis
Facebook API [44]	Ilimitado (1 p/ segundo)	Grátis

\*\*\*apenas eventos musicais

Por fim, para os eventos o número de APIs é um pouco limitado, no entanto, todas as analisadas não têm custos associados. A Eventful é uma das APIs mais completas, que está presente numa grande variedade de países e devolve dados de evento, como por exemplo: nome do evento, hora de início, hora de fim, localização geográfica, cidade/local e descrição. O serviço Bandsintown apenas devolve eventos musicais e o Facebook API sendo um serviço de uma rede social permite procurar eventos pelo nome e devolve informações como: descrição do evento, nome da localização e número de pessoas que responderam “Vou”, “Tenho interesse”, “Talvez” e “Não vou”.

## 2.4 Síntese

A secção descreve conceitos que são essenciais para uma melhor compreensão e planeamento do projeto proposto. Além disso esta pesquisa é importante para conhecer determinados protótipos já desenvolvidos, na área de *data warehouses* espaciais, que podem vir a ser essenciais na tomada de decisões no que diz respeito a funcionalidades e características que este projeto proposto pela NSN pode vir a ter. Além disto, o entendimento do papel que tem um servidor de mapas e os

padrões utilizados por este para a disponibilização de mapas são imprescindíveis para o cumprimento dos requisitos apresentados. Por último, a seleção de serviços que fornecem informação sobre contextos externos será essencial para que o cruzamento de informação da rede de telecomunicações com novos contextos de informação possa ser possível, de acordo com os requisitos do sistema apresentados na secção seguinte.



## 3 Requisitos do sistema

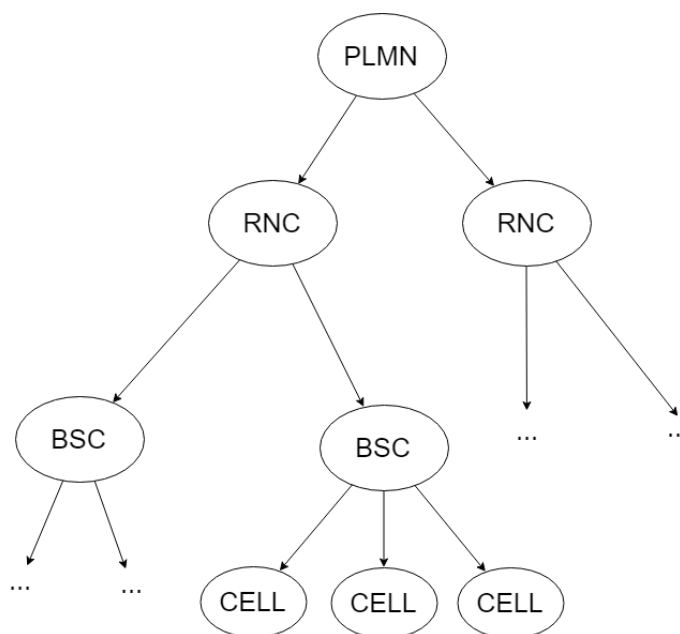
Esta secção apresenta o estado atual do sistema e conceitos sobre análise de desempenho em redes de telecomunicações, assim como conceitos necessários para a compreensão da solução proposta. Seguem-se os requisitos do sistema e as premissas propostas para o presente trabalho. Por fim, são analisadas as funcionalidades previstas de cada um dos módulos a desenvolver neste trabalho.

### 3.1 Sistema atual

As redes de telecomunicações atuais são sofisticadas, estando preparadas para trabalhar com tecnologias tão complexas como redes 3G, 4G e em breve 5G. Estas são compostas por variados elementos de rede com funções distintas, desde antenas físicas instaladas em postes de telecomunicações até elementos de rede virtualizados num computador. Porém, para o presente trabalho não é necessário um entendimento profundo na área das redes de telecomunicações, pois apenas são tratados os dados que estas fornecem ao sistema. É apenas necessário ter noção que cada um dos elementos de rede tem associado a si milhares de KPIs e que estes elementos pertencem a uma hierarquia, esquematizada na Figura 4. Os KPIs de cada elemento de rede são calculados com base em contadores que registam todo o tipo de métricas de cada antena, sendo possível, por exemplo, considerar o número de chamadas recebidas por uma antena, o número de SMS (*Short Message Service*) trocados, etc. Mais tarde, utilizando fórmulas e os valores destas métricas é feito o respetivo cálculo que define cada KPI.

A hierarquia de elementos de rede é composta por CELLS, na posição mais inferior, que são os elementos mais abundantes na rede. De uma forma simplificada, cada CELL corresponde a uma antena física de telecomunicações. As antenas pertencem a um determinado BSC (*base station controller*), que pode ser entendido como uma estação de antenas, que, por sua vez, pertencem a um RNC (*Radio Network Controller*), estando todos estes elementos associados a um PLMN (*Public*

*Land Mobile Network*). Estes dois últimos elementos mencionados (RNC e PLMN) são por norma elementos virtualizados computacionalmente.



*Figura 4 - Hierarquia de rede de telecomunicações*

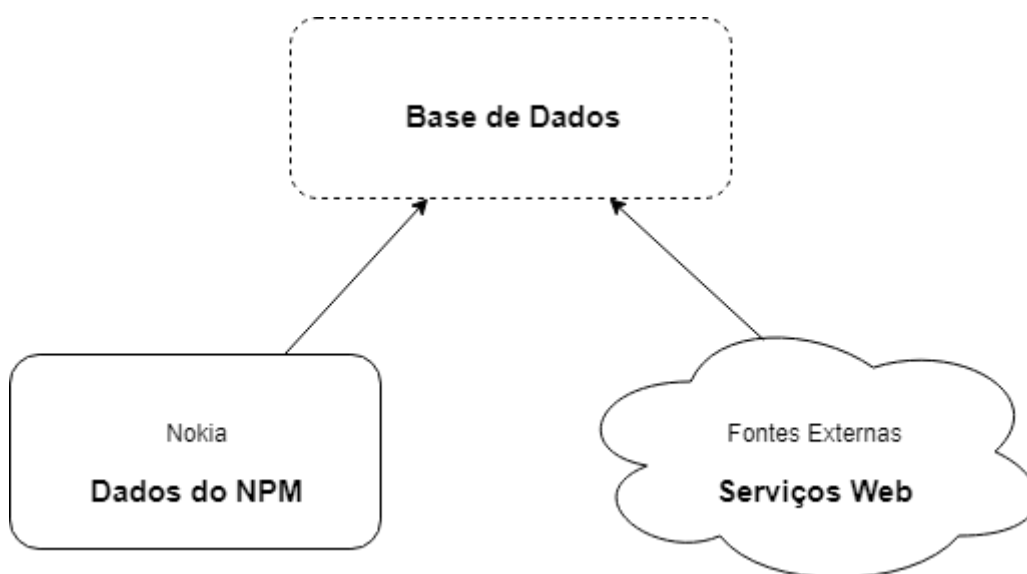
Todos os elementos de rede estão relacionados com outro elemento de um nível hierárquico superior na rede, à exceção dos PLMN. Todos os elementos de rede têm KPIs que medem a sua qualidade de funcionamento. Contudo apenas os CELLS têm associado uma localização geográfica e, por isso, apenas estes serão considerados para representação em mapas neste trabalho.

## 3.2 Requisitos

### 3.2.1 Requisitos funcionais

O trabalho em análise vem acrescentar contexto aos dados da ferramenta NPM. Isto é, além da informação disponibilizada pelas antenas e pelos KPIs calculados para cada elemento de rede, este trabalho vem acrescentar informação contextual, para que o analista perceba melhor o comportamento da rede de telecomunicações, podendo comparar o estado da rede com o estado do trânsito ou da meteorologia num determinado momento. O primeiro objetivo é perceber de que forma esta informação é obtida e tratada, e de que modo esta pode ser utilizada no NPM da NSN. Além disto, é necessário ter em atenção que existe informação que têm uma componente geográfica e que deve ser apresentada ao utilizador como tal para efeitos de análise e tomada de decisão.

Toda a informação relativa à rede de telecomunicações já é utilizada pela NSN, no entanto, a empresa não tem qualquer tipo de dados de outras fontes externas (meteorologia, trânsito, eventos) que possa utilizar para cruzar com a sua informação. Por isso, é necessário fazer uma pesquisa de fontes de dados externas com potencial para virem a ser usadas no NPM da NSN, e explorar de que forma a informação é fornecida e como deve ser tratada e filtrada para que possa vir a acrescentar valor ao sistema atual. Além disto, é necessário perceber de que modo os dados das fontes externas podem ser representados no mapa e como esta informação georreferenciada pode vir a ser usada no trabalho. Ou seja, a informação da rede pertence à ferramenta NPM da empresa, porém, os dados das fontes externas terão que ser obtidos utilizando serviços *web* já existentes (Figura 5).



*Figura 5 - Representação da origem dos dados*

Nem todos os dados fornecidos pela ferramenta NPM e obtidos das fontes externas são informação útil para o presente trabalho, principalmente os dados dos serviços *web*. Aquando da obtenção destes é necessário filtrar e tratar os dados, com o intuito de perceber quais os que realmente podem vir a ter influência na rede de telecomunicações e os que podem ser úteis para a representação final. Só depois deste processo de filtragem é que os dados devem ser armazenados tendo em conta a correlação entre eles e a forma como serão mais tarde disponibilizados e apresentados. Isto é, a título de exemplo, no caso dos dados meteorológicos, se estes forem obtidos por distritos e, no caso do trânsito automóvel, se este for obtido por troços de estrada, é possível correlacionar os distritos com as estradas e assumir que uma estrada pertence a um ou mais distritos. Além deste requisito, aquando do armazenamento da informação é necessário ter em conta que os dados com características geográficas têm de ser guardados como tal, para que seja possível que assim sejam disponibilizados e representados.

Depois de armazenada toda a informação necessária para ser apresentada, importa definir estratégias para disponibilizar essa informação na interface final. Estratégias estas que permitam não só tratar e disponibilizar os dados geográficos, como também os dados não geográficos associados a cada elemento georreferenciado.

Por fim, deverá ser encontrada a melhor forma de toda esta informação ser apresentada da melhor maneira, tendo em conta as suas características e usabilidade. Neste momento é essencial que a informação geográfica seja representada e que os dados não georreferenciados sejam acessíveis.

### **3.2.2 Requisitos não funcionais**

Quando se trata com uma solução de telecomunicações, como noutros sistemas, é necessário ter em conta os requisitos não funcionais de forma a que ausência destes não comprometa todo o sistema desenvolvido. Sendo este trabalho um estudo exploratório para um possível complemento ao portefólio da NSN, um dos requisitos não funcionais que terá mais importância neste ponto é o desempenho do sistema. Pretende-se que, o sistema desenvolvido seja interativo e não tenha atrasos significativos tanto na comunicação entre módulos como, e essencialmente, na apresentação de informação ao cliente. Para isto, se necessário, devem ser exploradas soluções para que o desempenho do sistema não seja um constrangimento na utilização do mesmo.

## **3.3 Funcionalidades dos módulos**

Com base nos requisitos do sistema abordados acima, é possível inferir que o presente trabalho é composto por quatro módulos principais apresentados na Figura 6: a base de dados (onde estes serão armazenados), o cliente dos serviços *web* (que terá a função de utilizar as APIs para obter os dados das fontes externas), a camada intermédia (que será responsável por tratar os dados geográficos e disponibilizá-los para a interface do cliente) e, por fim, a interface do cliente (que apresentará ao analista toda a informação necessária). Em seguida, nos tópicos subsequentes serão descritos com mais detalhe cada um destes módulos e as suas principais funcionalidades.

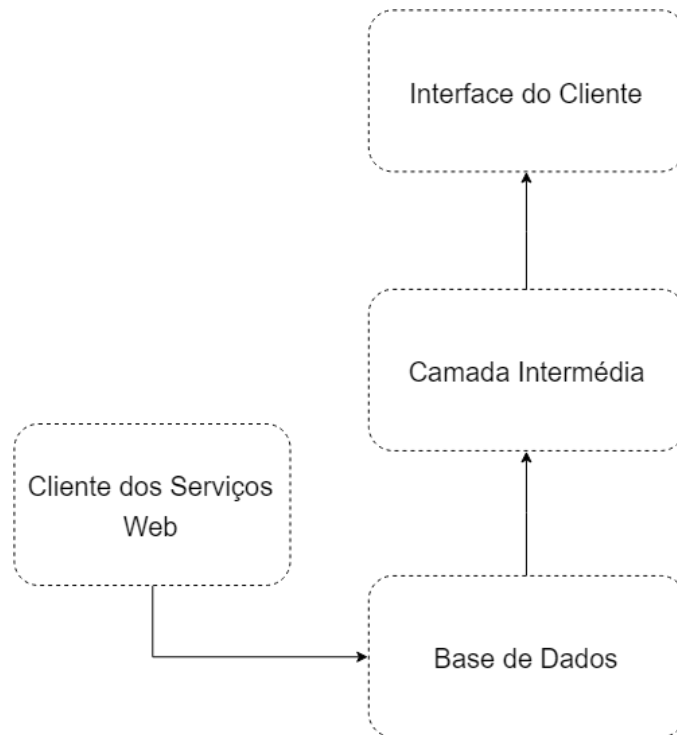


Figura 6 - Arquitetura conceptual do projeto

### 3.3.1 Base de dados

A base de dados tem de ser capaz de guardar uma grande quantidade de dados e também de lidar com dados georreferenciados que, dependendo do contexto das fontes externas, podem ser polígonos, linhas ou pontos geográficos. Além destas características, este módulo tem de ser capaz de comunicar com a camada intermédia, de forma a que os dados armazenados (geográficos e não geográficos) sejam reconhecidos e tratados tendo em conta o seu tipo, isto é, a título de exemplo: dados geográficos como informação a ser representada no mapa; data e hora como informação temporal do sistema.

A empresa NSN utiliza Oracle em todas as suas bases de dados e, por esta razão, se este *software* for capaz de corresponder a todos os requisitos apresentados acima, deverá ser a opção a escolher para este módulo.

### 3.3.2 Cliente dos serviços web

O cliente dos serviços *web* é o módulo responsável por obter e filtrar os dados das fontes externas e, de seguida, guardá-los na base de dados. É este que fará a chamada das APIs de cada um dos serviços escolhidos para fornecer informação de contexto e, em seguida, filtrar os dados obtidos para que a informação armazenada seja apenas a necessária para o problema. Por fim, para fazer a conexão à base de dados será utilizado o *driver* respetivo. Contudo, os dados das fontes externas têm

de ser recolhidos periodicamente, de hora a hora ou de dia em dia, dependendo da estratégia a adotar para cada contexto. Ou seja, além das tarefas referidas, este também tem de gerir o agendamento de processos para que estes pedidos cíclicos aconteçam.

De todas as APIs externas estudadas, o pedido de informação é feito utilizando um serviço *web* Hypertext Transfer Protocol (HTTP) REST e, sabendo que a linguagem de programação mais utilizada pela NSN é o Java, deve ser esta a escolhida para satisfazer os estes requisitos.

### 3.3.3 Camada intermédia

A camada intermédia é responsável por manipular a informação georreferenciada. Inicialmente, esta camada deve ser capaz de comunicar com a base de dados e conseguir distinguir informação geográfica de não geográfica. De seguida, deve ser capaz de tratar os objetos geográficos armazenados, criar uma camada de visualização (*layer*) para esses objetos num determinado momento temporal e posteriormente ter a capacidade de os disponibilizar utilizando protocolos específicos para fornecer este tipo de representação, como é apresentado na Figura 7.

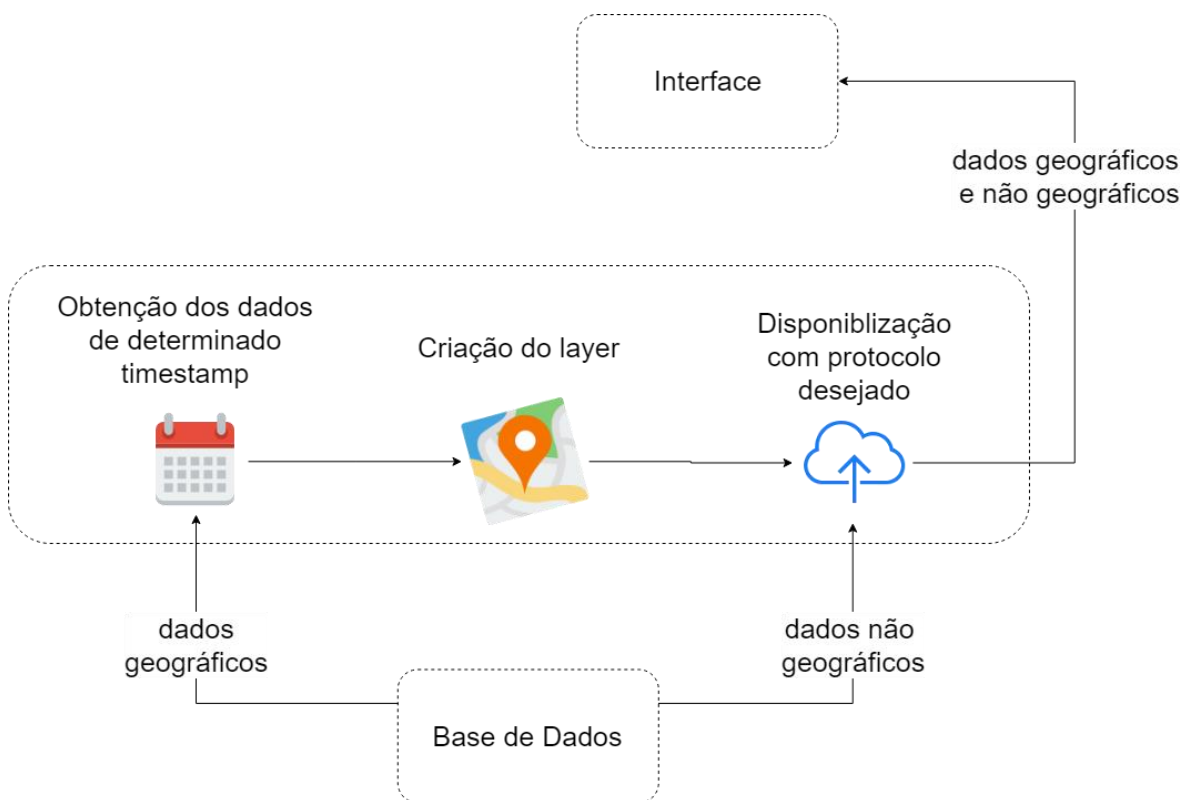


Figura 7 - Funcionamento da camada intermédia

### 3.3.4 Protótipo de demonstração

Por fim, a protótipo de demonstração (interface) é o módulo que comunica com o utilizador, apesar do foco do presente trabalho ser a forma como os dados são armazenados e disponibilizados, e não tanto como estes são apresentados. O sistema tem de ser capaz de disponibilizar toda a informação armazenada para o utilizador, mas sem grande nível de complexidade, apenas com alguns filtros onde seja possível controlar o que é apresentado no mapa.

Para a criação da interface além de HTML, CSS e da linguagem de programação JavaScript é necessário utilizar uma biblioteca capaz de representar mapas e *layers* utilizando os protocolos respetivos. Sabendo que a ferramenta NPM já utiliza este tipo de representação e que nesta é utilizada a biblioteca Leaflet, deve ser esta a escolhida para satisfazer os requisitos apresentados.

## 3.4 Atores e casos de uso

O utilizador alvo para o presente trabalho é um analista de uma operadora ou alguém com cargos de análise ou administração. Para o uso do sistema é necessário um conhecimento geral da rede de telecomunicações e um conhecimento aprofundado sobre os KPIs a serem analisados. Como já foi referido este trabalho tem como principal propósito o auxílio na interpretação de falhas que ocorreram no passado e a ajuda na previsão de falhas que possam vir a acontecer no futuro. Contudo, e para que esta deteção e previsão de falhas seja feita, é necessário que o analista que utiliza o sistema saiba exatamente que KPIs deve procurar e que contextos externos podem ter influência na rede de telecomunicações. Em contrapartida, o conhecimento tecnológico necessário para a utilização do sistema é nulo.

Tendo em conta que o presente trabalho foi criado como um estudo experimental em parceria com a empresa NSN, e reconhecendo que o foco do trabalho é o armazenamento e tratamento dos dados, a interface do utilizador é apenas representativa de como a informação pode ser apresentada e tem um nível de usabilidade bastante simples para qualquer utilizador. Este fator leva a que os casos de uso presentes para o trabalho em análise sejam também simples. Como podemos observar na Figura 8, o analista tem de ser capaz de escolher a data e hora da informação que quer analisar e conseguir ativar/desativar a representação no mapa dos valores de KPI, podendo de seguida escolher o KPI a ser apresentado. O mesmo fluxo lógico deste último é aplicado para as fontes externas, tendo de haver um filtro para cada contexto adicionado, por exemplo: se a estratégia é ter representados os valores de meteorologia, trânsito e eventos, tem de haver três filtros independentes para ativar e desativar cada um destes contextos, assim como escolher uma característica a ser apresentada.

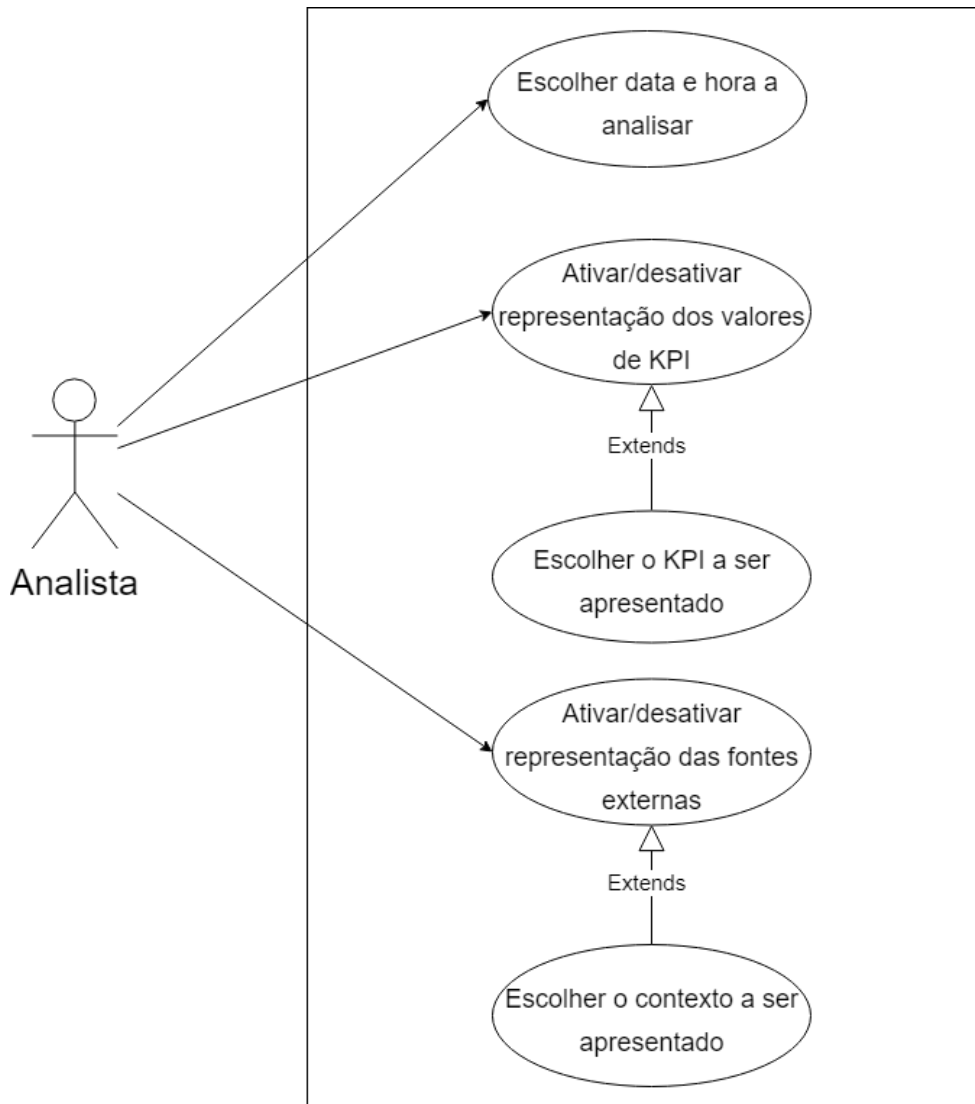


Figura 8 - Casos de uso do projeto

### 3.5 Síntese

Esta secção apresenta o estado atual do sistema, assim como, os requisitos para o projeto proposto, de forma a que os módulos da solução final começassem a ser delineados. Cada um destes módulos é expectável que desempenhe um papel diferenciado e objetivo no decorrer do trabalho, ajudando assim que a implementação, apresentada na secção seguinte, se torne faseada e com as funcionalidades a alcançar já previamente definidas. Por fim, de forma a que a implementação seguisse um propósito a ser alcançado, foram definidos os atores e os casos de uso a serem criados.



## 4 Arquitetura e implementação

Nesta secção é descrita a implementação de todo o sistema desenvolvido. Inicialmente é apresentada uma visão global do sistema, não entrando em detalhes de implementação, apenas apresentando os módulos que compõem a arquitetura do projeto. Nos tópicos subsequentes são descritos os módulos apresentados da referida arquitetura. Cada módulo descreve diferentes partes da mesma, sendo apresentados respeitando a ordem do fluxo de dados, servidor-cliente, e tendo sempre em consideração como é feita a comunicação entre cada um destes componentes.

### 4.1 Arquitetura do sistema

A arquitetura do sistema é composta por vários módulos distintos que comunicam entre si, como é possível verificar na Figura 9. Esta foi criada ao longo do tempo, tendo em consideração os desafios apresentados em cada fase de desenvolvimento do presente trabalho.

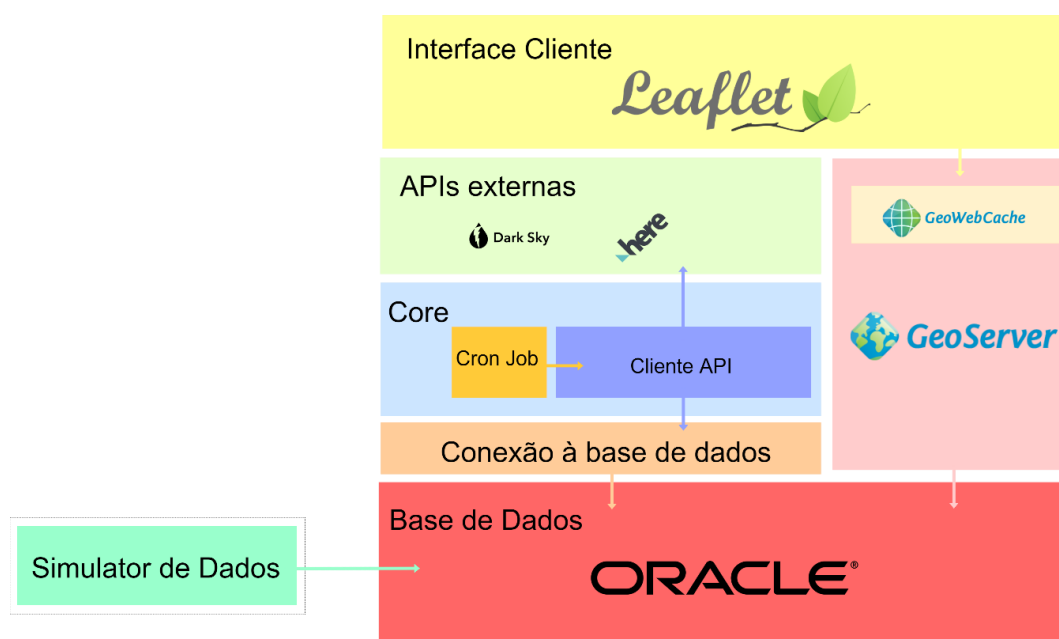


Figura 9 - Arquitetura do sistema

Para a camada da fonte de dados, tendo como base a topologia da rede e a estrutura dos dados correspondentes aos KPI da ferramenta NPM, foi necessário criar um esquema de base de dados que correspondesse a estes requisitos. No entanto, por razões de privacidade da informação da empresa, não foi possível ter acesso a dados de clientes e, para contornar este problema, foi criado um simulador da topologia da rede, assim como dos dados de KPI para cada elemento de rede. Desta forma, foi possível testar e representar esta informação, apesar de não serem valores reais da rede, mas apenas baseados nos mesmos.

Para a obtenção do contexto externo, foi feita uma análise de como obter a informação de cada API meteorológica, de trânsito e de eventos, e estudou-se que dados poderiam vir a ser úteis para este trabalho. Depois de filtrar estes dados e os adicionar à base de dados, foi criado um agendamento de processos para executar cada um dos pedidos de hora a hora ou diariamente, de forma a que fosse feita uma recolha de informação horária ou diária, respetivamente.

Tendo os dados disponíveis na base de dados, foi necessário criar uma camada intermédia que tratasse os dados geográficos e permitisse criar *layers* (camadas) de representação para mais tarde serem mostrados na interface do cliente utilizando o padrão WMS. Foi criada uma solução para que os *tiles* de cada *layer* fossem pré-calculados antes de serem pedidos pelo cliente e, desta forma, aumentar o desempenho, isto é, diminuir os tempos de resposta do sistema. Por fim, foi criada uma interface para o cliente onde toda a informação recolhida é mostrada e analisada.

Nos tópicos seguintes cada um destes módulos será descrito com maior detalhe assim como a comunicação que é feita entre eles.

## 4.2 Base de dados

A base de dados foi o primeiro módulo a ser criado não só por ser necessário criar um esquema eficiente para guardar a informação sobre a topologia de rede e os seus KPIs, como também para perceber a melhor solução para criar e guardar informação georreferenciada. A NSN atualmente usa o sistema de gestão de base de dados Oracle que, graças ao módulo Oracle Spatial Data Option, já é incluído em todas as edições recentes do Oracle [45]. Além disso, tem a capacidade de lidar com dados geográficos, assim como manipular pontos, linhas e polígonos num espaço 2D, sendo capaz de fazer operações geométricas com cada um deste tipo de dados. Este módulo disponibiliza um tipo de dados designado SDO\_GEOMETRY, que é o utilizado para armazenar objetos com características geográficas de qualquer tipo (ponto, linha, polígono) sendo este apenas uma abstração para os representar. Sabendo que as bases de dados Oracle são relacionais, e que o SDO\_GEOMETRY é um objeto representado como um tipo abstrato de dados, a informação de um objeto geográfico pode ser

guardada como um valor de um registo, desde que uma das colunas seja um objeto do tipo SDO\_GEOMETRY.

#### 4.2.1 Topologia da rede e indicadores de performance (KPIs)

Para a criação do esquema que guarda toda a informação associada à topologia de rede foi tida em atenção a hierarquia já apresentada dos elementos de rede e o facto de apenas os objetos CELL terem uma localização física. No entanto, todos os elementos de rede têm associados vários KPIs distintos que são calculados a partir de um conjunto de contadores usando fórmulas pré-definidas e armazenadas. Porém, tratando-se de KPIs não é correto aplicar, em todos os casos, agregações para determinar o valor de KPI de um elemento de rede que pertence a um nível superior na hierarquia. Isto é, a título de exemplo, o valor do KPI “A” para um BSC não pode ser calculado fazendo a média de todos os valores do KPI “A” dos CELLS que pertencem a este BSC. Este tipo de contagem que ser feito utilizando os contadores que são usados na fórmula de cada KPI. Consequentemente, estes contadores têm que ser adicionados à base de dados para que este tipo de cálculos seja possível ser efetuado.

A Figura 10 apresenta o esquema construído para que toda esta informação seja disponibilizada. Além de definir o tipo de objeto como SDO\_GEOMETRY também é necessário definir os metadados deste elemento como tal, só assim *softwares* como o Geoserver conseguem tratar de forma correta esta informação. Neste esquema cada um dos elementos da rede de telecomunicações é guardado na tabela “Network\_Objects”, a cada um destes está associado uma localização (“Network\_Objects\_Location”) definida por uma localização (SDO\_GEOMETRY) e *azimuth* (direção para a qual o elemento de rede está direcionado). Além disto, a cada um destes elementos também estão associados as métricas que os definem na tabela “NE\_COUNTER” e “COUNTER”, assim como os seus valores, e a data e hora a que valor corresponde. Por fim, em semelhança às métricas está também associado a cada elemento de rede os seus KPIs e a fórmula que define cada um destes.

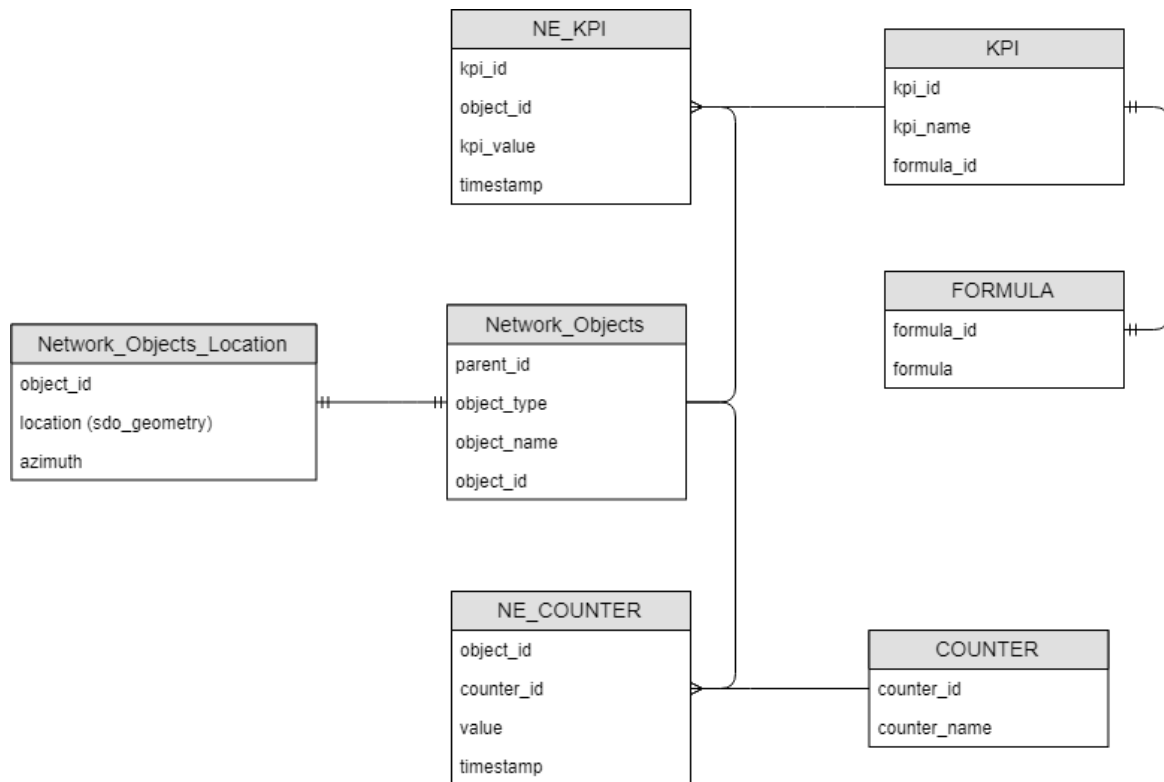


Figura 10 - Esquema da base de dados dos elementos de rede

#### 4.2.2 APIs externas

Para criar um esquema onde fosse guardada toda a informação necessária, foi definido à partida que seriam guardados três tipos de contexto, todos eles com características geográficas distintas: meteorologia, trânsito e eventos. A informação georreferenciada dos eventos é tratada da mesma forma que a localização dos elementos de rede, ou seja, um ponto no mapa. Já o trânsito e a meteorologia são abordados de forma diferente. No caso do trânsito cada um dos troços de estrada tem de ser guardado como uma linha, enquanto que no caso da meteorologia, foi assumido que a variação não é significativa dentro de um distrito e, como tal, a área a guardar é um polígono, mais concretamente os limites geográficos de cada distrito do país em análise. Todos estes contextos são criados como objetos do tipo SDO\_GEOMETRY, variando entre eles alguns parâmetros na sua criação, o número de coordenadas e os seus metadados. Para o caso da meteorologia, em que a área de cada polígono corresponde aos distritos do país, todos os dados recolhidos para inserir foram obtidos a partir da base de dados GDAM [46], que permite ter acesso a este tipo de informação para países, distritos ou concelhos. Estes dados foram obtidos no formato *shapefile* e em seguida adicionados à base de dados utilizando o *software* da Oracle: MapBuilder.

Um dos requisitos não funcionais para criar este esquema foi o facto de as localizações geográficas poderem estar relacionadas entre si, isto é, uma localização de um evento pertence a um

distrito, assim como uma estrada pertence a um ou mais distritos. Este tipo de relação podia ser solucionado de duas formas: tendo três esquemas independentes (um para a meteorologia, outro para o trânsito e outro para os eventos) em que, com recurso a operações do Oracle Spatial e com os SDO\_GEOMETRY de cada contexto, se verifique a interseção dos dados geográficos quando este tipo de relação for necessária; ou fazendo esta verificação no momento da inserção para todos os elementos e assim criar referências para os elementos com informação georreferenciada, sendo, conseqüentemente, os três esquemas independentes convertidos em apenas um. Como é preferencial ter mais desempenho no momento em que é feita uma *query* do que no momento da inserção dos dados, foi escolhido o esquema em que é pré-calculada a interseção entre os elementos geográficos, como é possível analisar no esquema da Figura 11.

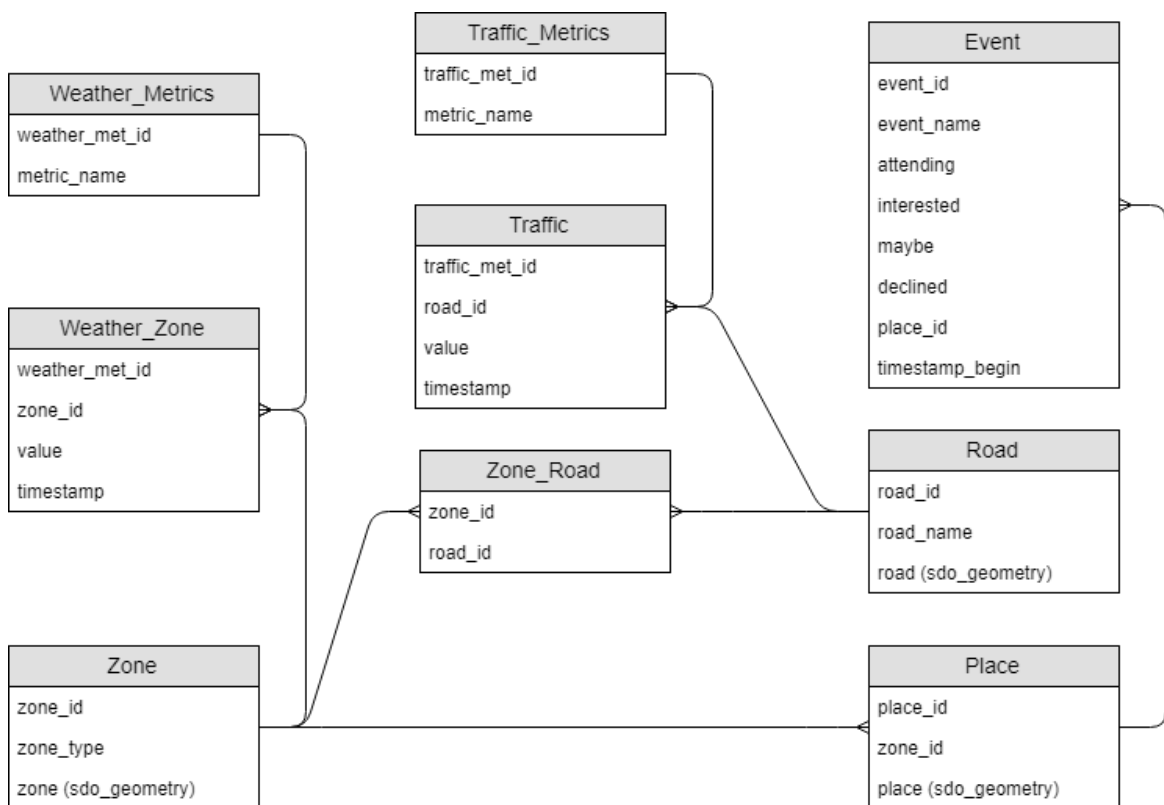


Figura 11 - Esquema de base de dados das fontes externas

Neste esquema os valores e o momento temporal de cada métrica meteorológica são guardados na tabela “Weather\_Zone”, incluindo o nome da métrica (exemplo: velocidade do vento, probabilidade de chuva, etc.) e a zona (distrito) a que este valor corresponde. No caso do trânsito a lógica é semelhante à meteorologia, sendo a localização um trecho de estrada (“ROAD”). Para os

eventos os dados apenas são guardados numa tabela (“Event”) que é definida pelo nome do evento, data e hora de início, número pessoas interessadas que vão tentar ir, que talvez vão e que não vão. A esta tabela está associada a localização do evento. Como referido acima as localizações estão todas associadas entre si, isto é, um troço de estrada pertence a um distrito, e um lugar (de um evento) pertence a um distrito. Toda esta verificação de interseção de localizações tem que ser efetuada no momento da inserção dos dados.

## 4.3 Simulador de dados

Por questões de privacidade e direitos de autor não foi possível utilizar os dados de clientes para criar a solução. Deste modo, foram criados dois *scripts* em Python que servem como acessório para a solução final, tendo apenas a função de preencher o esquema da topologia de rede e os valores de KPI com alguns dados simulados. Nos tópicos seguintes analisar-se-ão estes dois *scripts*: o primeiro para a criação da topologia de rede e o segundo para o preenchimento dos valores de KPI.

### 4.3.1 Topologia de rede

Para tornar os dados da topologia de rede o mais próximo da realidade possível, foi utilizada a informação das antenas da Vodafone [47], obtida num ficheiro CSV que descreve a localização das antenas de rede utilizadas pela operadora. Com esta informação obtém-se uma localização real da posição geográfica de cada antena que, na topologia de rede utilizada pela NSN, é caracterizada por CELL. O ficheiro tem dados de 29.395 antenas. Este *script* pode receber argumentos para o SQL ser escrito num ficheiro à parte ou diretamente na base de dados, tendo, para tal, de ser dado como valor de entrada a localização e credenciais da base de dados. Por fim, o caminho do ficheiro CSV com a descrição das antenas também é um dos argumentos obrigatórios. Na Figura 12 é apresentado o *workflow* deste *script*, começando por processar os argumentos obrigatórios para a sua execução, verificar se o utilizador pretende que o resultado seja escrito diretamente na base de dados ou num ficheiro à parte (fazendo a respetiva configuração à base de dados ou criação do ficheiro). Por fim, cria a topologia com base nas antenas da operadora e armazena-as respetivamente.

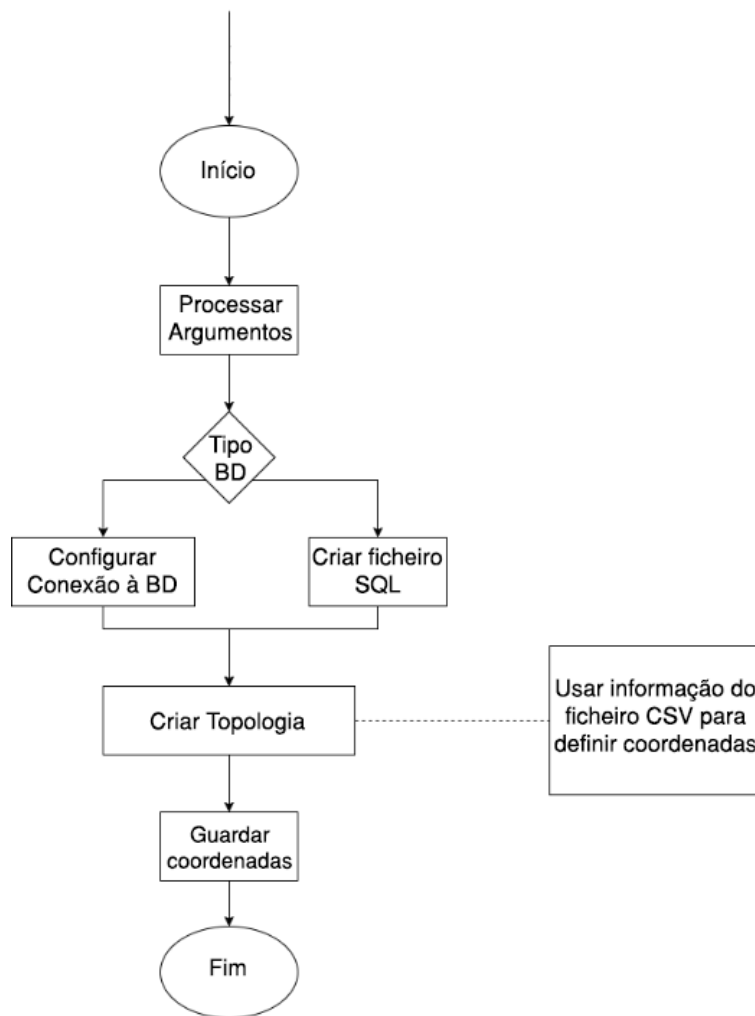


Figura 12 - Fluxo de processamento de script de criação de topologia de rede

#### 4.3.2 Valores dos indicadores de desempenho

Para o caso dos valores de KPI das antenas, apesar das regras privacidade da empresa e de não ser possível utilizar os dados dos clientes, seleccionaram-se sete dos KPIs mais importantes no que diz respeito a redes 3G e acedeu-se às suas fórmulas assim como ao seu valor mínimo, médio e máximo. No entanto, estes KPIs por serem dados privados da empresa não serão descritos neste relatório. Consequentemente, apesar de todos os valores criados para preencher a base de dados serem simulados, acabam por se aproximar da realidade com base no intervalo fornecido. Na Figura 13, pode ser observado o *workflow* deste *script* que também recebe como argumentos as credenciais da base de dados para a inserção dos dados ser direta e o nome de um ficheiro CSV com o descritivo e as fórmulas de cada KPI. O *script* começa por voltar a processar os argumentos, e decidir de que forma irá armazenar os dados criados. Em seguida, são simulados os valores de cada métrica e utilizadas as fórmulas respetivas de cada KPI para calcular o seu valor. Foi possível à NSN fornecer

o valor mínimo, máximo e médio de cada KPI, obrigando assim a que os dados, apesar de simulados, pertençam ao intervalo fornecido.

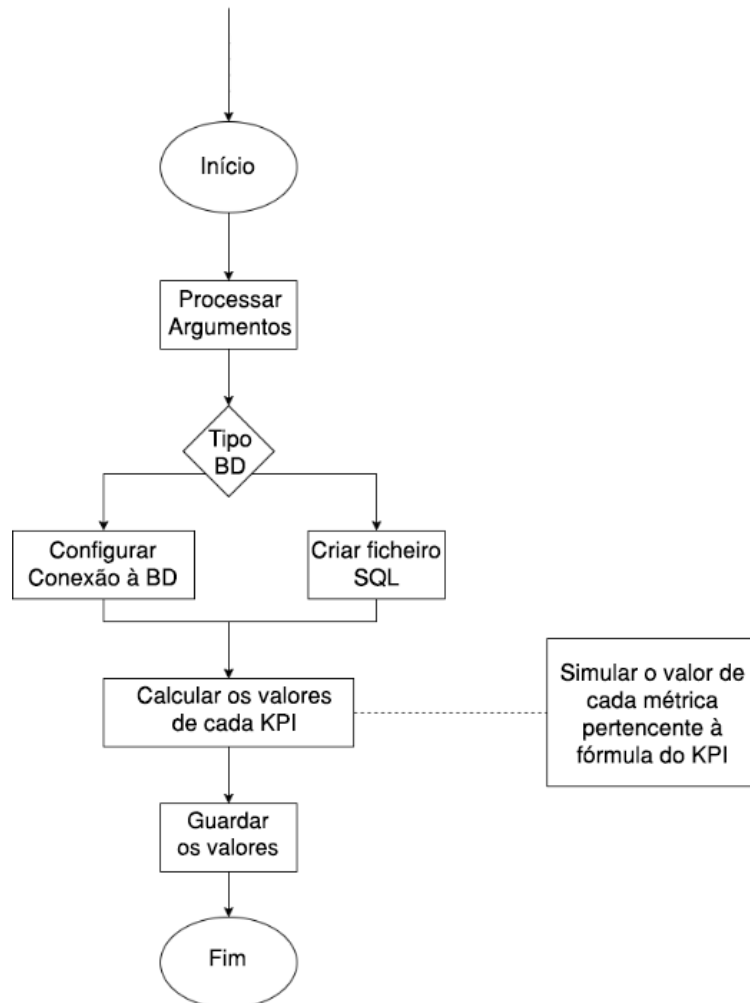


Figura 13 - Fluxo do processamento de script de simulador de dados

## 4.4 APIs externas

As APIs externas estão divididas em três grupos: meteorologia, trânsito e eventos, e para cada um destes contextos os pedidos de informação são feitos com pedidos HTTP GET para as respetivas plataformas. Para obter esta informação criou-se uma aplicação Java Maven em que foram feitos os pedidos e filtrados para obter os dados úteis para o presente trabalho. Para esta aplicação respeitaram-se vários padrões e bons hábitos de programação de forma a que trocar as API que fornecem cada um dos contextos fosse o mais fácil possível. No Apêndice 7 está representado o diagrama de classes desta aplicação. Neste esquema todos os contextos são tratados por classes diferentes, onde cada classe correspondente a uma API e é responsável por fazer o pedido dos dados e tratá-los.



#### 4.4.1 Meteorologia

No contexto meteorológico foi escolhida a DarkSky API, uma vez que cumpre os requisitos do presente trabalho e é isenta de custos até um dado limite de pedidos. No entanto, foi necessário recolher os dados da resposta ao pedido REST e selecionar os que poderiam ter maior impacto no desempenho da rede de telecomunicações. Como é visível na Figura 14, esta API oferece diversos dados meteorológicos, porém, para o problema em questão, apenas alguns deles podem vir a ter influência no contexto da rede. Com base nesta ideia os dados que foram escolhidos para aplicar a este estudo experimental foram: temperatura, velocidade do vento, probabilidade de precipitação, intensidade da precipitação e o ícone associado, este último porque dá uma visão rápida de como se encontra o estado do tempo numa determinada região, sendo útil para a representação dos dados na interface de cliente.

```
{
  "hourly": {"icon": "rain"...},
  "currently": {
    "ozone": 352.13,
    "windGust": 4.19,
    "temperature": 15.59,
    "icon": "partly-cloudy-day",
    "precipType": "rain",
    "humidity": 0.78,
    "cloudCover": 0.51,
    "summary": "Partly Cloudy",
    "apparentTemperature": 15.59,
    "pressure": 1016.09,
    "windSpeed": 3.34,
    "visibility": 10.01,
    "time": 1520939970,
    "windBearing": 206,
    "precipIntensity": 0.0305,
    "uvIndex": 3,
    "dewPoint": 11.82,
    "precipProbability": 0.03
  }, |
```

Figura 14 – Resultado do pedido da API Darksky

#### 4.4.2 Trânsito

Em relação ao tráfego automóvel foi utilizada a API Here Maps, visto que, entre as APIs estudadas para este propósito, esta é a única que fornece os dados em formato de texto. Nas restantes apenas é possível fazer o pedido dos *tiles* deste contexto, perdendo-se assim a possibilidade de saber o valor exato da qualidade de uma estrada. Apesar de ser um serviço pago, a Here tem disponível uma licença de avaliação gratuita por 90 dias para testar alguns dos seus serviços. Como é possível analisar na Figura 15, o resultado do pedido REST a este serviço não tem uma resposta clara da informação que é mostrada, para a qual a Here fornece uma documentação em que descodifica cada uma das siglas que são apresentadas. Desta resposta obtêm-se informações como: pontos geográficos que definem o troço de estrada em análise, qualidade da estrada, confiança na informação, código da estrada, nome, velocidade permitida e velocidade média que está a ser praticada no troço. No entanto,

para o estudo em questão apenas foram recolhidos os pontos geográficos, a qualidade, a confiança, o código e o nome da estrada, sendo que a informação mais importante tanto para a representação como para a análise é a qualidade da estrada. Este valor pode variar entre 0 e 10 e pode ser traduzido como fluxo de tráfego, isto é, por exemplo: uma estrada com o valor de 0 significa que o trânsito se encontra fluído enquanto que, por outro lado, se o valor for 8 ou 10 significa que esta se encontra muito congestionada ou parada, respetivamente.

```

{
  "CREATED_TIMESTAMP": "2018-02-23T16:19:53.000+0000",
  "UNITS": "metric",
  "VERSION": "3.2",
  "RWS": [
    {
      "RW": [
        {"PBT": "2018-02-23T16:19:33Z"...},
        {"PBT": "2018-02-23T16:19:03Z"...},
        {"PBT": "2018-02-23T16:19:53Z"...},
        {
          "FIS": [...],
          "PBT": "2018-02-23T16:19:04Z",
          "DE": "N235",
          "mid": "09e373f0-0987-4d56-baeb-026eab46e1b0|",
          "LI": "842-11481"
        },
        {
          "FIS": [...],
          "PBT": "2018-02-23T16:19:09Z",
          "DE": "N235",
          "mid": "fed11a07-4057-4572-a716-b5a4d12a6058|",
          "LI": "842+11481"
        },
        {
          "FIS": [
            {...}
          ],
          "PBT": "2018-02-23T16:19:05Z",
          "DE": "N335",
          "mid": "4e29fe7e-ab99-4c48-b9aa-a4c060c85709|",
          "LI": "842-12804"
        },
        {
          "FIS": [...],
          "PBT": "2018-02-23T16:19:10Z",
          "DE": "N335",
          "mid": "4a614aca-f60e-4bd0-b4a6-bcc4c40952b5|",
          "LI": "842+12804"
        }
      ],
      "TY": "TMC",
      "EXTENDED_COUNTRY_CODE": "E4",
      "TABLE_ID": "42",
      "UNITS": "metric",
      "EBU_COUNTRY_CODE": "8",
      "MAP_VERSION": "201801"
    },
    "MAP_VERSION": ""
  ]
}

```

Figura 15 - Resultado do pedido da API Here

#### 4.4.3 Eventos

No que aos eventos concerne, não foi possível obter um serviço que preenchesse os requisitos necessários para esta tarefa. Inicialmente a solução para este contexto passou por utilizar a API

Eventful que disponibiliza informações como: nome do evento, local, ponto geográfico, descrição e hora de início e término. Contudo, apenas com este serviço não era perceptível qualquer tipo de informação sobre a popularidade de um evento, ou o número previsível de pessoas que participariam no evento. Para contornar este problema a solução passava por utilizar a Facebook API, para fazer uma correspondência entre eventos e assim saber a quantidade de pessoas que confirmaram a sua presença e as que confirmaram a apenas possibilidade de ir a determinado evento (que corresponde à resposta «Talvez» nos eventos do Facebook). De salientar que esta estratégia passava sempre por procurar um evento em primeiro lugar no serviço Eventful e só depois fazer a correspondência na API do Facebook, nunca de forma inversa já que esta é uma rede social onde qualquer pessoa registada pode criar um evento público, havendo grande diversidade de eventos que podem ser classificados como falsos positivos, uma vez que os utilizadores têm liberdade total para criar eventos (muitas vezes com motivações políticas ou sociais) que não vão existir.

No entanto, esta estratégia delineada para obter os dados para este contexto não pôde ser implementada porque, aquando dos testes da API Eventful, detetou-se que as localizações geográficas dos eventos registados neste serviço não correspondiam à localização real do evento. A título de exemplo, um evento cujo nome e restantes dados estavam corretos, tinha como localização geográfica Lisboa e no local do evento o Coliseu dos Recreios, quando, na realidade, este se realizaria em Coimbra, como pode ser inferido na Figura 16. A acrescer a esta situação, em março de 2018 gerou-se uma polémica à volta da empresa Facebook sobre a falta de privacidade dos dados dos seus utilizadores, o que fez com que a sua API deixasse de estar disponível devido a alterações que seriam efetuadas nas políticas de privacidade da empresa. Tendo em consideração que dos dados recolhidos os mais importantes seriam a localização de um evento a par do número de pessoas que o evento iria abarcar, a estratégia inicialmente elaborada ficou sem efeito.

```
venue_name:      "Coliseu dos Recreios"
country_name:    "Portugal"
watching_count:  null
country_abbr2:  "PT"
description:     " <strong>Promotor</strong><p>Universidade de Coimbra - Teatro Académico de Gil Vicente<strong>Sinopse</strong><p>A história entre María Nieves Rego e Juan Carlos Copes, dois famosos bailarinos na história do tango. María e Juan conheceram-se quando tinham 14 e 17 anos respetivamente, e dançaram juntos durante cerca de 50 anos. Durante todo esse tempo, amaram-se e odiaram-se, passaram por várias separações e voltaram a juntar-se. Neste documentário, María e Juan partilham a sua história com um grupo de jovens bailarinos de tango de Buenos Aires, que transformam os momentos da sua vida em incríveis coreografias. Uma viagem inesquecível ao coração do tango.<strong>Realizador</strong><p>German Kral<strong>Preços</strong><p><br>Plateia - 4€<strong>Descontos</strong><p><br>Alliance Française<br>Cartão Fnac<br>Comunidade UC<br>Desempregado<br>Estudante<br>Grupo = ou > 10 pax<br>Maiores de 65 anos<br>Menores de 25 anos<br>Ord. Engenh. Reg. Centro<br>Rede UC</p></p></p></p>"
start_time:     "2018-03-05 21:30:00"
```

Figura 16 - Resultado do pedido da API Eventful

Por fim, não obtendo uma API capaz de substituir os dois serviços acima referidos, a opção de mostrar os eventos foi abandonada. Todavia, o sistema descrito até este ponto está preparado para que este tipo de informação seja adicionada a qualquer momento. Para isto basta inserir uma API que corresponda às necessidades anteriormente referidas, substituindo no diagrama apresentado no Apêndice 7 a classe correspondente ao serviço do Facebook ou Eventful e fazendo as necessárias alterações nos requisitos pretendidos.

## **4.5 Cliente das APIs e *agendamento de processos***

Uma das premissas para o presente trabalho corresponde ao facto de este dever guardar a informação proveniente das APIs externas de hora em hora ou diariamente, dependendo do contexto em questão. Para satisfazer este requisito e depois de ter a recolha e a filtragem dos dados necessários para a análise que se pretende, é fundamental ter um cliente para estes serviços externos que faça o controlo da informação que estes tratam, isto é, que faça o pedido da informação e que garanta que estes são guardados na base de dados. Além deste cliente que gere os acessos à informação das APIs também é necessário ter um agendamento de processos (*cron jobs*) responsáveis por executar este gestor de hora em hora ou diariamente.

### **4.5.1 Cliente das APIs externas**

Como anteriormente referido o cliente das APIs externas é o módulo que controla quando são feitos os pedidos a cada API e quando a informação proveniente dos mesmos é guardada. Este módulo também pertence à aplicação Java Maven que contém o acesso às APIs externas que foi descrito no tópico 4.4 e desta forma comunica diretamente com os serviços escolhidos para fornecer os dados contextuais. O funcionamento deste módulo é bastante simples: ele faz a comunicação direta com o módulo das APIs externas para que este recolha os dados que se pretende guardar, e com o módulo que comunica com a base de dados para fornecer os dados a guardar. Na Figura 17 pode ser analisado o diagrama de classes para este módulo, que também inclui uma classe com o nome de TileCaching, que é responsável por comunicar com o servidor de *cache* para pré-calculas as *tiles* da informação que é guardada na base de dados. Contudo, este assunto será abordado com mais detalhe no tópico 4.6 deste relatório.

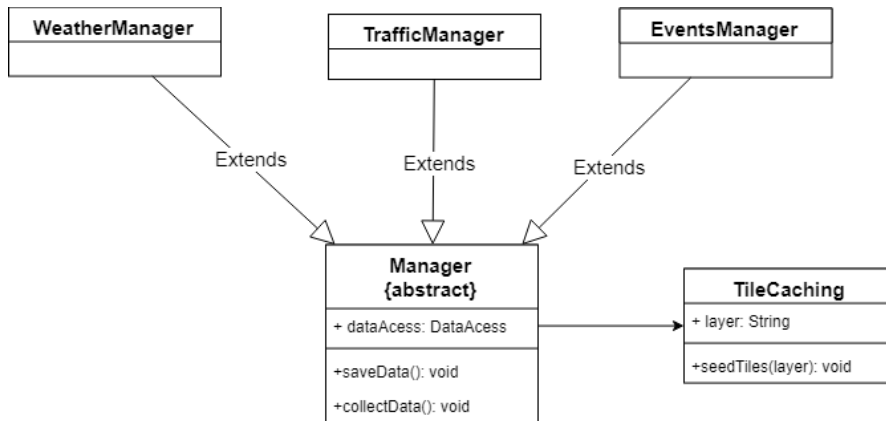


Figura 17 - Diagrama de classes do cliente das APIs

#### 4.5.2 Conexão à base de dados

A conexão à base de dados é outro módulo que pertence à aplicação Java Maven, e comunica diretamente e apenas com o cliente das APIs externas, descrito acima. Este módulo tem o propósito de escrever os dados dos serviços externos na base de dados. Atendendo a que a base de dados escolhida para este projeto foi o Oracle, naturalmente este módulo utiliza o *driver* Oracle JDBC (*Java Database Connectivity*) para fazer a comunicação com a mesma. Como pode ser observado no diagrama de classes da Figura 18, é utilizado o driver Oracle, no entanto pode ser facilmente alterado para uma base de dados distinta criando a respetiva estrutura de dados que utilize o *driver* desejado e implementando a *interface* DatabaseDriver.



Figura 18 - Diagrama de classes da conexão à base de dados

#### 4.5.3 Agendamento de processos

Recordando que um dos requisitos deste estudo é o facto de a recolha de informação contextual ser feita periodicamente de hora em hora ou de dia em dia, foi necessário criar um módulo que tratasse de executar o cliente das APIs de forma periódica. Para tal, com recurso à biblioteca Quartz [48], que permite executar código periodicamente e calendarizar processos, foi criado este módulo que possibilita a recolha cíclica da informação meteorológica, de trânsito e dos eventos.

Tanto a informação meteorológica como a de trânsito são recolhidas de hora em hora. No caso da meteorologia, este processo é feito para todos os distritos que estiverem armazenados, enquanto que no trânsito, a recolha é feita para todas as estradas que o serviço de trânsito fornecer. No caso dos eventos, apesar de não ter sido implementado o serviço para fornecer tal informação, seria executado de forma diária, guardando os eventos dos próximos 30 dias e, atualizando todos os dias, se necessário, os eventos futuros.

Em termos técnicos este módulo é composto por um gestor de *cron jobs* que gere todas as especificações dos processos a serem executados periodicamente, por exemplo: horário a ser praticado, regularidade, classe que será executada. Por outro lado, existe uma classe CronJob, como é possível observar na Figura 19, que tem de implementar uma interface da biblioteca Quartz para que seja reconhecido como um *cron job* e é este processo que será executado periodicamente. Para o caso de estudo em análise é este processo que comunica diretamente com o cliente das APIs.

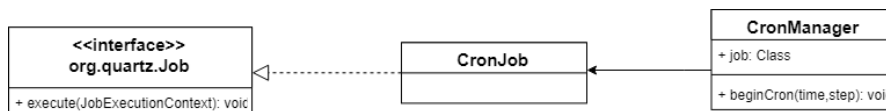


Figura 19 - Diagrama de classes do agendamento de processos

#### 4.5.4 Comunicação entre módulos

Sumariando, todos os módulos abordados nos tópicos anteriores (APIs externas, cliente das APIs externas, conexão à base de dados e os *cron jobs*) pertencem à mesma aplicação Java Maven e a sua comunicação é feita de forma direta. No entanto, esta comunicação não é feita entre todas as classes e de forma desorganizada. Todos os dados seguem um fluxo lógico e é com base neste fluxo que os módulos comunicam entre si. Os *cron jobs* executam o cliente das APIs externas de forma periódica. Este cliente comunica com o módulo das APIs externas, que fazem o pedido HTTP GET dos dados e filtra-os. Em seguida quando o módulo das APIs externas devolve estes dados filtrados ao cliente das APIs, este comunica com o módulo que é responsável pela conexão à base de dados para que estes sejam guardados. Na Figura 20 é possível ter uma visão global de como todas as classes comunicam entre si - dentro de cada módulo encontra-se o diagrama de classes que foi apresentado anteriormente.

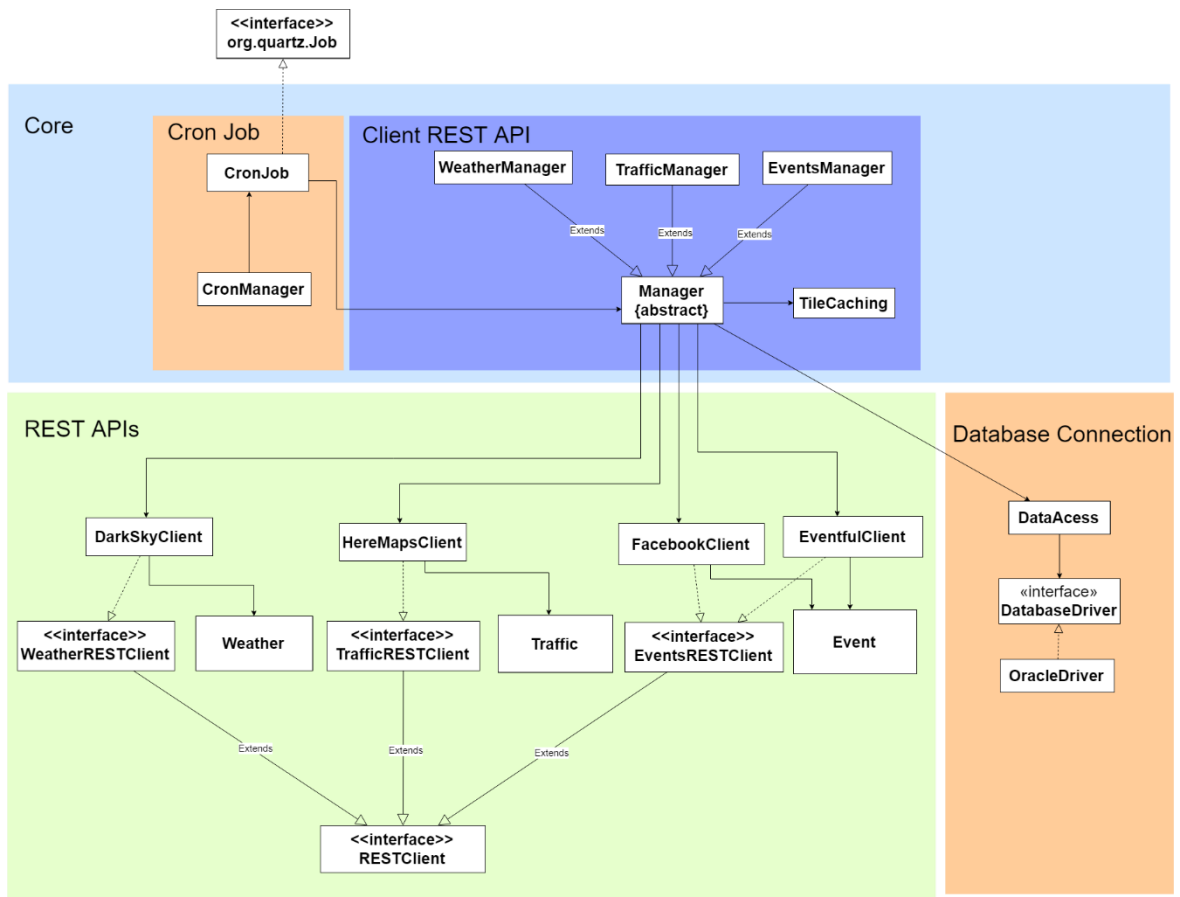


Figura 20 - Diagrama da comunicação entre módulos

## 4.6 Geoserver e GeoWebCache

Neste tópico é descrito o funcionamento da camada intermédia no fluxo de informação, entre a base de dados e a interface de cliente. É explicada a necessidade e a importância desta camada, nomeadamente da plataforma Geoserver para este efeito. Em seguida, são apresentadas as razões pelas quais é necessário um *software* para pré-calculer cada camada criada no Geoserver e descreve-se a plataforma que foi utilizada para este propósito, o GeoWebCache. Este módulo do ponto de vista do trabalho efetuado é dos mais importantes a par das capacidades do Oracle Spatial, sendo aqui tratados os dados geográficos como tal e resolvido um dos problemas abordados na secção 2, que tem que ver com a frequência com que os dados geográficos estão presentes na base de dados de uma empresa, mas que acabam por ser tratados como texto (moradas) ou números (latitude, longitude).

Foi selecionada a plataforma Geoserver uma vez que esta, a par com o MapServer, são capazes de lidar com os dados geográficos e disponibilizá-los como *layers* para a camada superior. No entanto, o que distingue uma da outra é que o Geoserver usa a linguagem Java e o MapServer o

C/C++, o que levou a que o Geoserver fosse a escolhida porque a NSN desenvolve os seus projetos em Java.

#### 4.6.1 Geoserver

O Geoserver é responsável por tratar toda a informação georreferenciada que está armazenada na base de dados. Esta informação tem de estar mapeada como SDO\_GEOMETRY para o caso do *software* Oracle. Além da informação georreferenciada o Geoserver também é capaz de levar para o cliente outro tipo de dados da base de dados que estejam associados a um objeto geográfico. Este serviço pode ser instalado de várias formas e em múltiplas plataformas, mas no caso em estudo foi executado como uma *webapp* que corre num *application server*, o Tomcat. Para a sua configuração este *software* tem disponível uma interface, como é apresentada na Figura 21, em que podem ser criadas todas as camadas e configuradas todas as funcionalidades do sistema.

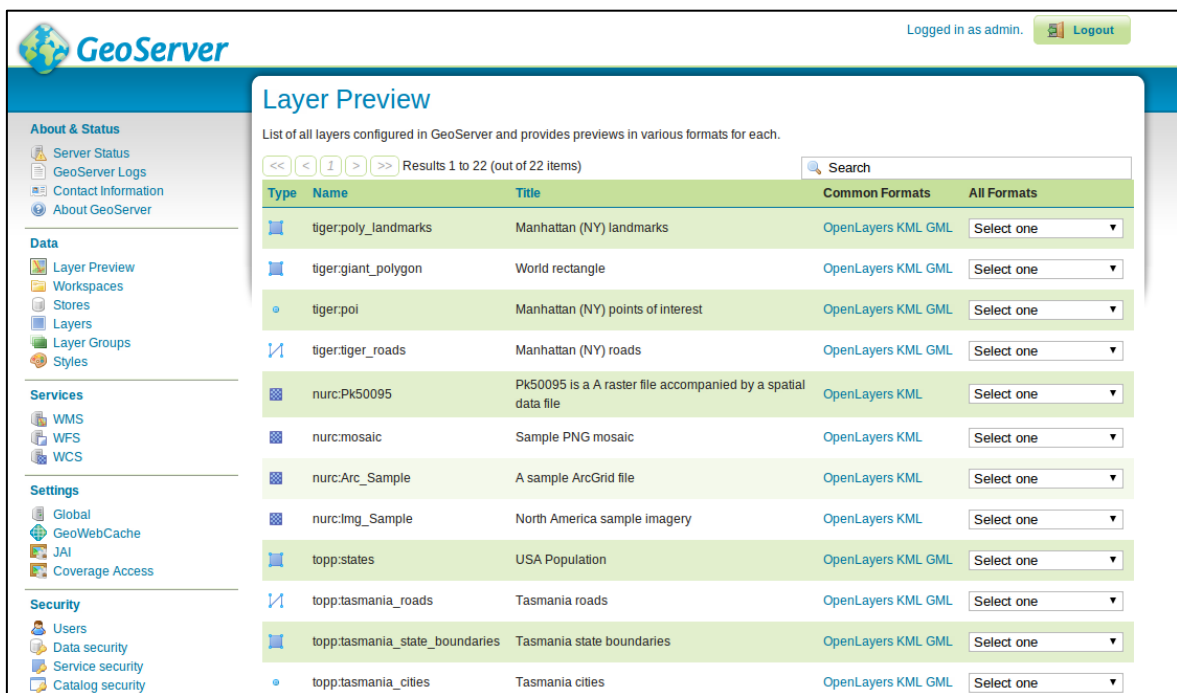


Figura 21 - Interface do Geoserver

Antes de qualquer configuração, tem que ser instalada uma extensão para ser possível a comunicação com bases de dados Oracle. Esta extensão tem por base o *driver* Oracle JDBC porque o Geoserver é uma aplicação Java. Todas as instalações de qualquer tipo de extensões e *plugins* são simples e estão bem documentadas no site Geoserver. Posteriormente, é necessário ter em consideração que há três tipos de camadas de dados: *workspace*, *store* e *layer*. O *workspace* é uma camada que não é obrigatório criar quando se pretende configurar *stores* ou *layers*, mas tem o propósito de organizar a informação existente no Geoserver. A *store* tem de pertencer sempre a um



*workspace* e é esta camada que faz a comunicação com a base de dados. Para isto na sua criação é necessário configurar os detalhes de comunicação com a base de dados como é apresentado na Figura 22.

The screenshot displays the GeoServer web interface for configuring a new vector data source. The left sidebar contains navigation menus for 'About & Status', 'Data', 'Services', 'Settings', 'Tile Caching', 'Security', 'Demos', and 'Tools'. The main content area is titled 'New Vector Data Source' and includes the following sections:

- Basic Store Info:** Includes a dropdown for 'Workspace \*' set to 'nokia', a text input for 'Data Source Name \*', and a text input for 'Description'.
- Connection Parameters:** Includes text inputs for 'host' (localhost), 'port' (1521), 'database \*', 'schema', 'user \*' (admin), and 'passwd'. It also features a checked checkbox for 'Enabled', a 'Namespace \*' dropdown set to 'http://geoserver.org/nokia', an unchecked checkbox for 'Expose primary keys', and text inputs for 'max connections' (10), 'min connections' (1), and 'fetch size' (1000).

Figura 22 - Configuração da conexão à base de dados no Geoserver

Por fim, depois do *workspace* e a *store* configurados podem ser criados os *layers* para representação de informação georreferenciada. Os *layers* podem ser criados a partir de uma tabela que esteja presente na base de dados ou criando uma vista SQL. Nesta vista é formulada uma consulta em SQL que mostre os dados que se quer apresentar e a informação que é necessária ter associada em cada *layer*. Depois dos dados escolhidos é necessário configurar os seguintes parâmetros: o sistema de referência das coordenadas (no caso em estudo foi usado o EPSG:4326), a área de visualização (*bounding box*) em que os dados serão representado na *layer* a criar (depende sempre dos dados de análise), as dimensões (para o caso de existir algum tipo de *timestamp*, o que para o estudo em análise é fundamental tendo em conta que os dados guardados são distintos de hora a hora) e, por fim, o estilo de cada *layer*. Este último é o que permite um ponto ou um polígono serem representados como tal no mapa, isto é, a forma, cor ou estilo de um elemento geográfico varia

conforme o estilo escolhido. O Geoserver também permite criar novos estilos que, utilizando a linguagem Styled Layer Descriptor (SLD) baseada em linguagem Extensible Markup Language (XML), permite que sejam criadas regras para que a representação de cada elemento tenha determinados filtros, condições ou premissas. A título de exemplo, no caso em estudo do trânsito, a qualidade da estrada é definida por um número de 0 a 10 e a representação de uma estrada e o seu fluxo de trânsito é feita com uma linha de uma determinada cor. Assim pode ser considerado que para determinado intervalo de valores essa linha seja pintada de verde (para retratar o trânsito fluído) e que para outro intervalo de valores essa linha seja pintada de vermelho (para retratar o trânsito congestionado). Este tipo de detalhes são funcionalidades que podem ser configurados em cada estilo. A Figura 23 apresenta um pequeno exemplo em SLD que representa todos os pontos de um determinado *layer* com um círculo vermelho de tamanho 6.

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <StyledLayerDescriptor version="1.0.0"
3   xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd"
4   xmlns="http://www.opengis.net/sld"
5   xmlns:ogc="http://www.opengis.net/ogc"
6   xmlns:xlink="http://www.w3.org/1999/xlink"
7   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
8   <NamedLayer>
9     <Name>Simple point</Name>
10    <UserStyle>
11      <Title>GeoServer SLD Cook Book: Simple point</Title>
12      <FeatureTypeStyle>
13        <Rule>
14          <PointSymbolizer>
15            <Graphic>
16              <Mark>
17                <WellKnownName>circle</WellKnownName>
18                <Fill>
19                  <CssParameter name="fill">#FF0000</CssParameter>
20                </Fill>
21              </Mark>
22              <Size>6</Size>
23            </Graphic>
24          </PointSymbolizer>
25        </Rule>
26      </FeatureTypeStyle>
27    </UserStyle>
28  </NamedLayer>
29 </StyledLayerDescriptor>
```

Figura 23 - Exemplo de estilo, retirada de [49]

Por fim, tendo em conta os dados disponíveis, foram criados três *layers* para representar os dados de trânsito, com base nos valores da qualidade da estrada: trânsito fluído, (com valores entre

0 e 2), trânsito pouco fluído (com valores entre 2 e 5) e trânsito congestionado (com valores entre 5 e 10). Na Figura 24 está representado *layer* criado para o trânsito fluído do dia 2 de maio de 2018 às 16 horas.

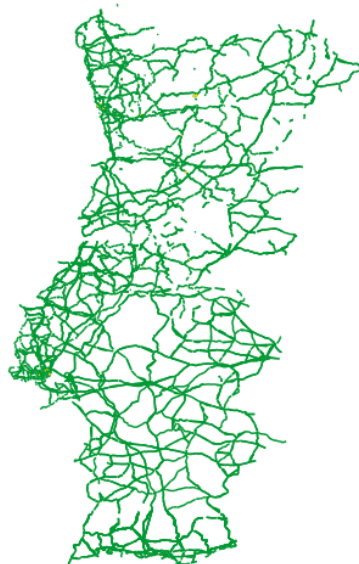


Figura 24 - *Layer* do trânsito fluído do dia 2 de maio de 2018 às 16 horas

Para o caso da meteorologia foram criados dez *layers* distintos, diferenciados pelo ícone que representa cada estado de tempo, sendo estes: dia limpo, noite limpa, dia pouco nublado, noite pouco nublada, nublado, chuva, granizo, neve, vento e nevoeiro. Cada ícone está representado na Figura 25 pela referida/respetiva ordem.



Figura 25 - *Ícones da meteorologia*

Para o contexto da topologia de rede foram criados sete *layers*, o número que KPIs disponibilizados pela empresa, sendo estes representados no mapa com as cores vermelho, amarelo e verde conforme o seu valor de KPI, como é apresentado na Figura 26.

Todos os *layers* criados para cada contexto têm o fundo transparente e aquando da representação de vários *layers* na mesma visualização estes são sobrepostos uns aos outros, sem nunca se perder informação porque todas as localizações são distintas umas às outras.

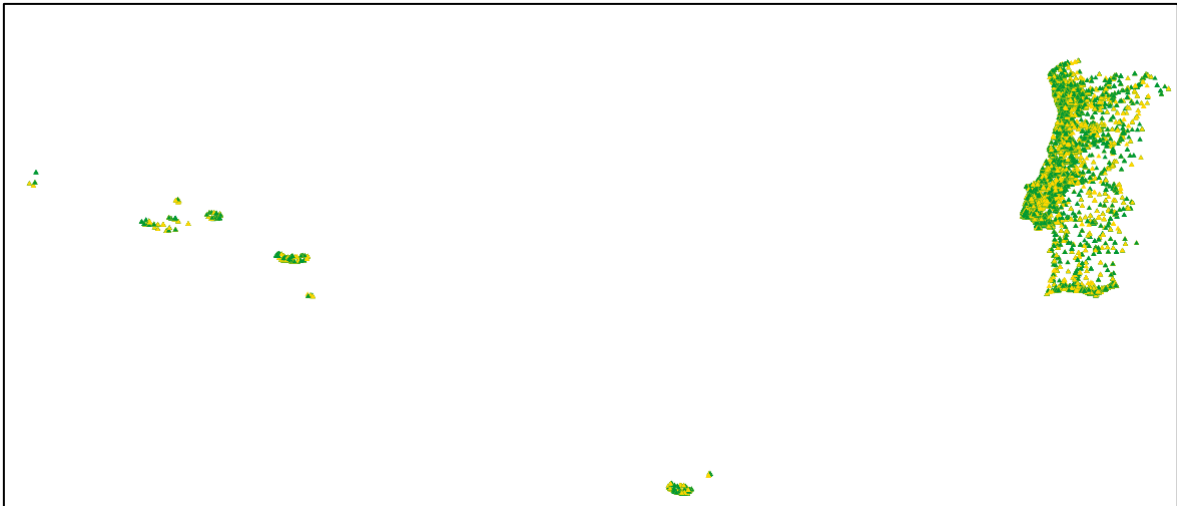


Figura 26 - Exemplo de um layer KPI

A partir deste momento todos os *layers* depois de criados já estão disponíveis para ser consumidos, utilizando o protocolo WMS, com recurso à ligação <http://localhost:8080/geoserver/wms>, para o caso de o Geoserver se encontrar em *localhost* no porto 8080.

#### 4.6.2 GeoWebCache

O protocolo escolhido para fornecer a informação de cada *layer* foi o WMS, porque segundo a análise feita no tópico 2.2, este é mais eficiente que o WFS para representar grandes quantidades de dados. O WMS devolve os *tiles* da camada pedida, neste caso em formato PNG (Portable Network Graphics). Porém, o seu grande problema de desempenho baseia-se no facto de, aquando da realização do pedido por parte de um cliente, o servidor de mapas (Geoserver) calcular e criar os *tiles* para a *layer* pedida e só depois disto é que estes são enviados. Ou seja, quando este pedido é feito para um *layer* com muitos objetos geográficos vai demorar mais tempo a ser calculado, e se o mesmo pedido voltar a ser solicitado ao Geoserver, este vai ter que o voltar a calcular. Para contornar esta questão pensou-se numa solução a fim de criar *cache* para cada *layer* ou, se possível, pré-calcular e guardar os *tiles* antes destes serem pedidos pelo cliente. De entre as soluções existentes, a escolhida para este propósito foi o *software* GeoWebCache, isto porque tem um *plugin* de integração direta com o Geoserver. Depois deste *plugin* instalado as configurações ao GeoWebCache podem ser feitas utilizando a *interface* do Geoserver. Para adicionar um *layer* ao GeoWebCache basta permitir que a *cache* dos *tiles* seja efetuada na página de cada *layer*, sendo, de seguida, necessário configurar alguns parâmetros (formato das imagens dos *tiles*, tempo de expiração de cada *tile*, filtros, entre outros) como pode ser observado na Figura 27.

The screenshot shows the 'Tile Caching' configuration tab for a layer. It includes several sections:

- Title cache configuration:**
  - Create a cached layer for this layer
  - Enable tile caching for this layer
  - Enable In Memory Caching for this Layer.
  - BlobStore: (\*) Default BlobStore
  - Metatiling factors: 4 tiles wide by 4 tiles high
  - Gutter size in pixels: 0
  - Tile Image Formats:
    - application/json;type=utfgrid
    - image/gif
    - image/jpeg
    - image/png
    - image/png8
    - image/vnd.jpeg-png
  - Expire server cache after n seconds (set to 0 to use source setting): 0
  - Expire client cache after n seconds (set to 0 to use server setting): 0
- Parameter Filters:**
  - TIME
  - Default String: 2018-05-01T16:00:00.0Z
  - Allowed Strings (one per line):
    - 2018-05-01T00:00:00.0Z
    - 2018-05-01T01:00:00.0Z
    - 2018-05-01T02:00:00.0Z
    - 2018-05-01T03:00:00.0Z
    - 2018-05-01T04:00:00.0Z
    - 2018-05-01T05:00:00.0Z
    - 2018-05-01T06:00:00.0Z
    - 2018-05-01T07:00:00.0Z
    - 2018-05-01T08:00:00.0Z
  - Case: Case Sensitive
  - Locale (For case conversion):
- STYLES:**
  - Default Style: LAYER DEFAULT
  - Alternate Styles:  ALL STYLES
  - Add filter: Choose One
  - Add Style filter:
- Gridset:**
  - Gridset: EPSG:4326
  - Published zoom levels: Min / Max
  - Cached zoom levels: Min / Max
  - Add grid subset: Choose One

Figura 27 - Configuração do cache de um layer

De seguida, ao encontro de um requisito não funcional proposto, o GeoWebCache permite que os *tiles* em *cache* sejam guardados numa base de dados (*disk quota*), no qual é possível configurar a localização e acessos à base de dados. Além disto, também é possível controlar o limite de espaço em disco que os *tiles* podem ocupar, obrigando o GeoWebCache a apagar os *tiles* menos utilizados quando este limite é atingido. Na Figura 28, é possível analisar estas configurações.

### Disk Quota

Enable disk quota

Disk quota check frequency:  
 Seconds  
 (Last run: 4 s ago.)

Maximum tile cache size  
 GiB

Using 668.0 KB of a maximum 200.0 GB

When enforcing disk quota limits, remove tiles that are:  
 Least frequently used  
 Least recently used

Disk quota store type

Target database type

JDBC data source

JDBC Driver class name

JDBC connection URL

User name

Password

Min. connections

Max. connections

Connection time out (ms)

Validation query

Max open prepared statements

Figura 28 - Configuração da base de dados do GeoWebCache

Depois destas configurações efetuadas, os *tiles* já podem ser acedidos pelo GeoWebCache, neste caso através de `http://localhost:8080/geoserver/gwc/service/wms` no caso do Geoserver se encontrar em *localhost* no porto 8080. Aquando da realização dos pedidos ao GeoWebCache, este devolve de imediato os *tiles* solicitados se estes se encontrarem em *cache*, caso contrário, retorna o pedido para o Geoserver e devolve o resultado, guardando os *tiles* em falta na *cache*. Porém, esta solução não resolve a possibilidade de ter os *layers* pré-calculados antes mesmo de um cliente fazer o pedido. Para tal, o GeoWebCache tem disponível um serviço REST que permite que, com uma resposta a um comando HTTP POST (em formato XML ou JSON), os *tiles* pedidos sejam carregados para *cache* a qualquer momento. Nesta resposta os parâmetros que têm de ser configurados são: o nome do *layer*, SRS, o intervalo de *zoom*, o formato da imagem, o tipo de operação, entre outros (Figura 29).

```

<?xml version="1.0" encoding="UTF-8"?>
<seedRequest>
  <name>topp:states</name>
  <bounds>
    <coords>
      <double>-2495667.977678598</double>
      <double>-2223677.196231552</double>
      <double>3291070.6104286816</double>
      <double>959189.3312465074</double>
    </coords>
  </bounds>

  <!-- These are listed on http://localhost:8080/geowebcache/demo -->
  <gridSetId>EPSG:2163</gridSetId>
  <zoomStart>0</zoomStart>
  <!-- This will be reduced to 3, since the layer is only defined for 0-3 -->
  <zoomStop>2</zoomStop>
  <format>image/png</format>

  <!-- type can be * seed (add tiles) * reseed (replace tiles) * truncate (remove tiles) -->
  <type>truncate</type>

  <!-- Number of seeding threads to run in parallel.
       If type == truncate only one thread will be used regardless of this parameter -->
  <threadCount>1</threadCount>
  <!-- Parameter filters -->
  <parameters>
    <entry>
      <string>STYLES</string>
      <string>pophatch</string>
    </entry>
    <entry>
      <string>CQL_FILTER</string>
      <string>TOTPOP > 10000</string>
    </entry>
  </parameters>
</seedRequest>

```

Figura 29 - Pedido POST para pré-cálculo de layer

Para que seja possível retirar proveito desta funcionalidade, isto é, para que os *tiles* sejam pré-calculados, o cliente das APIs, depois de pedir a informação aos serviços utilizados e depois de a guardar na base de dados, através da classe *TileCaching*, faz o pedido HTTP POST para que os *tiles* dos contextos guardados sejam pré-calculados antes de serem pedidos por qualquer cliente.

## 4.7 Interface do cliente

A interface do cliente está organizada de forma simples, tendo apenas o intuito de testar todos os *layers* criados no Geoserver, dando a possibilidade de criar alguns filtros e tirar conclusões. Notar que, o foco principal deste trabalho é a forma como os dados contextuais são tratados,

guardados e disponibilizados até ao cliente, tendo menos importância a sua interface. Saliente-se porém, que todos os dados guardados têm de ser acessíveis nesta última.

Para aceder aos *layers* criados no Geoserver, foi necessário ter uma biblioteca Javascript que permitisse que os pedidos WMS fossem feitos. Para tal, de entre as opções analisadas a *framework* Leaflet foi a escolhida, sendo que para o protocolo WMS os pedidos são efetuados de forma simples, como pode ser constatado na Figura 30.

```
var wmsLayer = L.tileLayer.wms('https://demo.boundlessgeo.com/geoserver/ows?', {  
  layers: 'nasa:bluemarble'  
}).addTo(map);
```

Figura 30 - Pedido WMS no Leaflet, retirado de [50]

A interface tem que apresentar a informação de cada *layer*, criado no Geoserver, ao cliente. No entanto, se esta apresentar toda a informação ao mesmo tempo, o cliente não consegue fazer qualquer tipo de análise devido ao número de camadas apresentadas. Para resolver este problema as vistas do utilizador podem ser alteradas seguindo os seguintes filtros: data e hora, KPI, estado do tempo e estado do trânsito a analisar, filtros estes que podem ser observados na Figura 31 (o nome dos KPI foram propositadamente ocultados).



Figura 31 - Filtros da interface final

Em seguida, para fornecer os *tiles* dos mapas recorreu-se ao serviço de mapas da Here por este ser utilizado para fornecer os dados de trânsito, havendo assim o mínimo de erros no que diz respeito à correspondência de cada estrada com o mapa. Na Figura 32 pode ser observada esta interface que tem características bastante simples e intuitivas para qualquer utilizador.



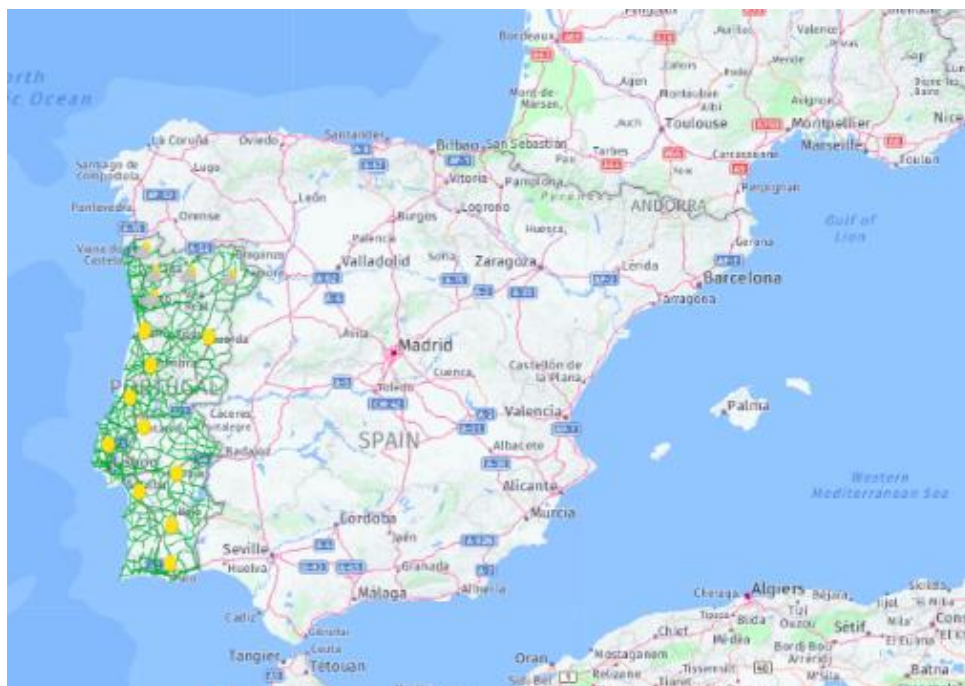


Figura 32 – Recorte da apresentação final da interface

Por fim, devido às novas capacidades do protocolo WMS é possível seleccionar um elemento geográfico, mostrando este a devida informação associada ao mesmo e, desta forma, ter acesso a toda a informação criada no SQL *view* do Geoserver aquando da criação de cada *layer*. Na Figura 33 é apresentado um exemplo desta capacidade do protocolo referido.

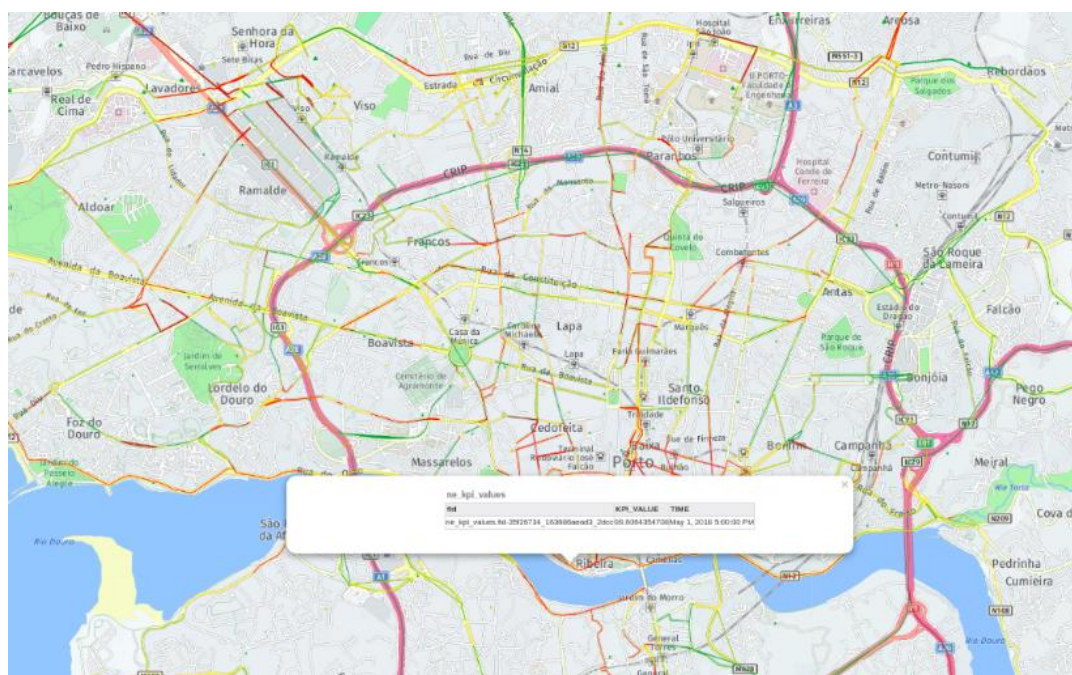


Figura 33 - Seleção de elemento

## 4.8 Síntese

Ao longo da fase de implementação, as funcionalidades de cada módulo foram ganhando forma e enquanto umas decisões no desenvolvimento do projeto foram tomadas previamente, antes da implementação começar (exemplo: escolha da base de dados), outras foram tomadas aquando do desenvolvimento, por exemplo: o servidor de mapas, o padrão de disponibilização de mapas, a criação de *cache*. Sendo este um projeto com vários módulos e tecnologias diferentes não houve tempo para que cada uma das destas decisões fossem comparadas e testadas (exemplo: escolha de MapServer ou Geoserver), por isso todas estas opções foram baseadas nas pesquisas apresentadas na secção 2. Apesar de este ser um protótipo de uma solução e de existir trabalho futuro bem definido, todo o trabalho proposto foi alcançado, e todos os módulos definidos nos requisitos foram desenvolvidos com sucesso, o que tornou possível que os testes e resultados, apresentados na secção seguinte, pudessem ser efetuados sem que algum módulo fosse simulado.

## 5 Testes e resultados

Esta secção está dividida em dois tópicos distintos: funcionalidade e desempenho. No primeiro são descritos o ambiente de testes, os dados utilizados e duas histórias de uso de utilizadores (*user stories*). No segundo são descritas estratégias para aumentar o desempenho, tempo e espaço de armazenamento do pré-cálculo de um *layer*, uma estimativa de espaço de armazenamento necessário ao longo do tempo e tempos de apresentação de um *layer* com e sem *cache*.

### 5.1 Funcionalidade

#### 5.1.1 Ambiente de testes e dados utilizados

Para o desenvolvimento do presente trabalho foram utilizadas duas máquinas para alojar dois dos módulos: a base de dados Oracle e o Geoserver, sendo ambas semelhantes e tendo as seguintes características:

- Processador CPU: Intel Xeon CPU E5-2680 v3 @ 2.50GHz
- Memória RAM: 16Gb
- Capacidade de armazenamento: 500Gb

O Geoserver é executado no servidor de aplicações TomCat e foram dedicados 12Gb de memória RAM para o seu funcionamento.

Para testar o sistema considerou-se a área geográfica de Portugal, sendo que os dados georreferenciados utilizados para a topologia de rede se obtiveram de acordo com as antenas da Vodafone. Para o caso meteorológico, em que a área de cada polígono corresponde a distritos do país, os dados recolhidos também foram de Portugal continental e insular. No entanto, para o trânsito, a API escolhida apenas tem os dados de Portugal continental e, por esta razão, apenas estes foram utilizados (Figura 34).

Por outro lado, em relação aos valores de cada um dos contextos, no caso dos KPIs, foram criados dados simulados baseados em intervalos reais. Em contrapartida, tanto para os valores meteorológicos como para os valores de trânsito estes foram fornecidos pela respetiva API, ou seja, todos eles são valores reais.

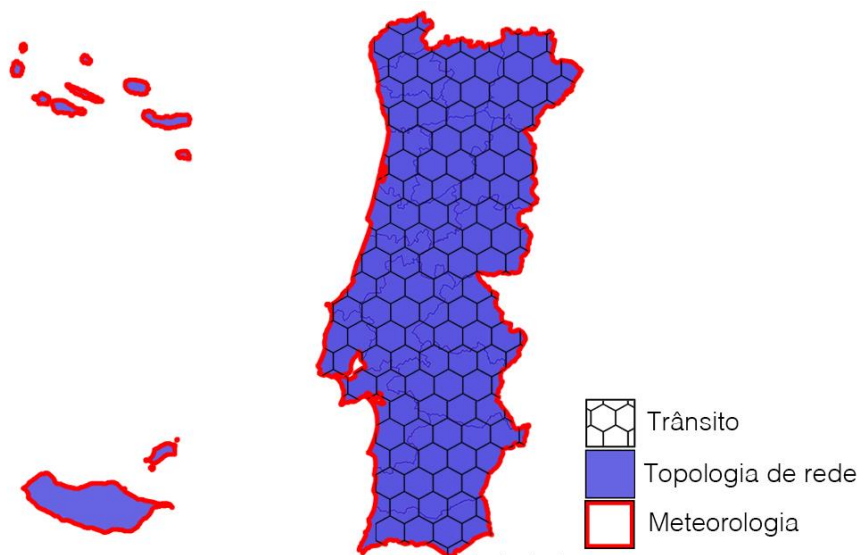


Figura 34 - Cobertura geográfica dos diferentes contextos

### 5.1.2 Histórias de uso

Uma das características dos sistemas de análise de rede atuais tem que ver com o facto de estes apenas devolverem a informação da rede, não podendo esta ser cruzada com outro tipo de dados para perceber algum tipo de comportamento da rede face a outros contextos. Este conceito vai ao encontro da motivação do presente trabalho, vindo este ajudar à compreensão da rede por parte do analista e fornecer informação extra sobre o ambiente em que a rede de telecomunicações se encontra. Assim, enquanto analista, é possível perceber se a sobrecarga de uma antena se pode relacionar com um congestionamento de trânsito numa área geográfica próxima, ou se determinadas condições meteorológicas afetaram os valores normais dos KPIs. Com base nestes casos de uso, o sistema atual consegue dar resposta às histórias de uso descritas em seguida.

- Como analista quero saber quais as condições meteorológicas e de trânsito, no dia 30/2/2018 às 17h, para perceber se a falha de rede junto da Torre dos Clérigos pode estar relacionada com algumas destas situações.
- Como analista quero saber se a tempestade em Lisboa, no dia 24/12/2017 às 23h, fez com que alguma antena deixasse de funcionar corretamente; esta informação pode ser obtida se o valor do KPI\_123\_A (nome de KPI fictício) for superior a 35.

## 5.2 Persistência

Um dos requisitos para o presente trabalho foi que este permitisse apresentar dados para análise do desempenho das redes com o tempo de resposta, optando-se, por esta razão, por pré-calcular os *tiles* antes dos pedidos do cliente serem efetuados em vez de o fazer no momento em que é feito cada pedido. Outra estratégia para aumentar o nível de desempenho foi aumentar o número de *layers* pré-calculados. Ou seja, o Geoserver permite criar um *layer* que receba como argumento um valor enviado pelo cliente no momento de cada pedido, sendo esta a primeira abordagem que foi utilizada neste trabalho. No entanto, para gerar o pré-cálculo de um *layer* que recebe argumentos é necessário criar todas as opções que este *layer* pode receber. Por exemplo, o caso do trânsito tem como filtros: trânsito fluído, trânsito pouco fluído e trânsito congestionado. Para pré-calcular este *layer*, que recebe como argumento o estado do trânsito a apresentar, é necessário ter armazenado situações onde estão os três filtros visíveis, onde estão apenas dois ou onde só está visível um deles, isto é, 7 visualizações distintas. Por esta razão, a estratégia escolhida foi criar um *layer* diferente para cada filtro, sendo assim armazenadas apenas 3 visualizações (para o exemplo acima) e cada uma delas é apresentada quando o utilizador o pretender sem interferir nas restantes.

A grande contrapartida de ter uma solução de pré-cálculo de *tiles* é o espaço de armazenamento que esta solução pode requerer ao longo do tempo. Uma das particularidades a definir quando é feito este pré-cálculo é o nível de *zoom* a armazenar, podendo este intervalo ser essencial para o bom funcionamento do sistema, isto é, se o intervalo de nível de *zoom* armazenado for reduzido, o espaço de armazenamento ocupado também o é, mas este nível de *zoom* será demasiado reduzido para ser possível fazer qualquer tipo de análise. No entanto, por outro lado, se o intervalo de *zoom* for demasiado grande o espaço de armazenamento ocupado não é sustentável. Na Tabela 7 são apresentados o número de *tiles*, o tempo de criação e o espaço ocupado para quatro intervalos de *zoom* distintos para o *layer* do trânsito fluído do dia 1 de maio de 2018 às 16 horas, no formato PNG. Como pode ser analisado o número de *tiles* e o espaço ocupado, de nível para nível, cresce de forma exponencial, chegando mesmo a passar 10 Tb de espaço ocupado. Em apêndice estão alguns exemplos do mapa com níveis de zoom diferentes (Apêndice 1, Apêndice 2, Apêndice 3, Apêndice 4, Apêndice 5, Apêndice 6).

Tabela 7 - Características dos diferentes níveis de zoom

Nível de zoom	Número de tiles	Tempo de criação	Espaço ocupado
1 - 10	1 154	< 10 segundos	6,34 Mb
1 - 12	12 674	≈ 1,5 minutos	60,52 Mb
1 - 15	736 498	≈ 1 hora	2,92 Gb
1 - 21	2 994 226 018	indeterminado	11,6 Tb*

\*Valor estimado com base no nível de zoom anterior

O Geoserver permite definir o formato dos *tiles* guardados. Neste estudo foi utilizado o formato PNG, no entanto, o Geoserver oferece outros formatos que podem ser mais vantajosos que o formato utilizado, como JPEG (Joint Photographic Experts Group), GIF (Graphics Interchange Format) e PNG8. Na Tabela 8 é apresentado o espaço de armazenamento que outros formatos ocupam em comparação com o PNG. Estes valores foram obtidos utilizando o *layer* de trânsito fluído do dia 1 de maio de 2018 às 16 horas, com nível de zoom 1-12. Como é possível concluir o formato utilizado é o que ocupa mais espaço de armazenamento, apesar da diferença não ser significativa para este nível de zoom, a longo prazo e em diferentes níveis de zoom utilizar o formato GIF pode vir a ser uma mais valia.

Tabela 8 - Espaço ocupado por diferentes formatos de tiles

Formato	Espaço de armazenamento
PNG	60,52 Mb
JPEG	50,91 Mb
PNG8	48,03 Mb
GIF	45,99 Mb

Além do espaço de armazenamento que o pré-cálculo dos *tiles* ocupa também é necessário considerar o espaço de armazenamento que os dados ocupam na base de dados. Apesar deste último não se aproximar do espaço de armazenamento que tem de ser dedicado para a persistência dos *tiles*, ao longo do tempo pode ser significativo. Como é apresentado na Figura 35, os valores para um dia de armazenamento da topologia de rede e dos KPIs ocupam cerca de 196Mb, enquanto que para as fontes externas ocupam 73Mb, 72Mb destes para o trânsito e apenas 1Mb para armazenar a informação meteorológica.

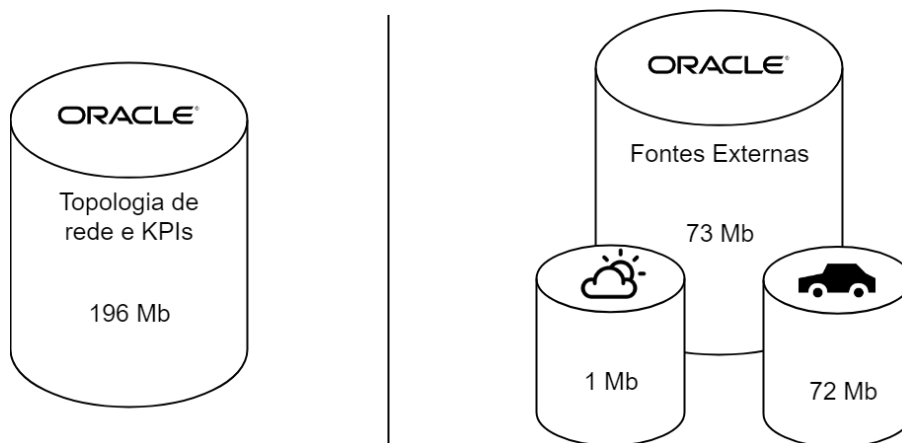


Figura 35 - Armazenamento diário na base de dados

Com base nos valores apresentados, na Tabela 9 é estimado o espaço de armazenamento necessário para esta solução durante um determinado intervalo de tempo. Na coluna da base de dados é apresentado o espaço de armazenamento que os esquemas da mesma ocupariam se os valores diários se mantivessem semelhantes, tanto para o esquema da topologia de rede e KPIs como para o esquema das fontes externas. Já na coluna “GeoWebCache” é apresentado o espaço que o pré-cálculo dos *tiles* poderá ocupar desde 1 dia até 1 ano. De referir que o valor acumulado de rede e KPIs engloba todos os valores dos 7 KPIs utilizados para o presente trabalho, o acumulado meteorológico engloba os 10 filtros que este contexto pode ter e o acumulado de trânsito engloba os 3 filtros escolhidos. Os valores meteorológicos e de KPIs são muito maiores que os valores de trânsito, porque a área de cobertura de cada um é diferente (Figura 34). Enquanto que para o trânsito apenas é considerado Portugal continental, nos restantes é considerado Portugal continental e insular. E apesar da área das ilhas dos Açores e da Madeira serem reduzidas e não acrescentarem muito mais informação, todos os *tiles* da *bounding box* que define estes *layers* são guardados, até mesmo os *tiles* transparentes, sendo, por isso, tão discrepante a diferença no espaço de armazenamento ocupado.

Tabela 9 - Espaço de armazenamento de difentes intervalos de tempo

Duração	Base de Dados			GeoWebCache ( nível de zoom: 1-12)			
	Rede e KPIs	Fontes externas	Acumulado	Acum. Rede e KPIs	Acum. Meteo.	Acum. Trânsito	Acumulado
1 dia	196 Mb	73 Mb	269 Mb	106,1 Gb	144 Gb	4,4 Gb	254,5 Gb
1 semana	1,3 Gb	511 Mb	1,8 Gb	742,6 Gb	1 Tb	30,8 Gb	1,77 Tb
1 mês	5,9 Gb	2 Gb	7,9 Gb	3,2 Tb	4,2 Tb	132 Gb	7,53 Tb
1 ano	71,5 Gb	26 Gb	97,5 Gb	38,7 Tb	51,3 Tb	1,6 Tb	91,6 Tb

Os valores apresentados acima, principalmente do GeoWebCache, são apenas uma estimativa do espaço de armazenamento necessário durante um determinado intervalo de tempo. No entanto, na prática, o que normalmente acontece é que os relatórios mais acedidos são os das últimas 24 horas, enquanto que os restantes são solicitados com menos frequência, não se justificando, por esta razão, ter *tiles* pré-calculadas de há 1 semana ou mais. No GeoWebCache tem que ser definido o espaço de armazenamento utilizado para guardar *tiles*, espaço este que pode ser configurado a qualquer altura e, quando o limite deste armazenamento é atingido, o gestor de *cache* apaga os *tiles* mais antigos para guardar nova informação. Tendo em conta esta funcionalidade, no presente trabalho o espaço de armazenamento definido para que esta gestão dos *tiles* fosse efetuada, foi de 254,5 Gb, com base nos valores obtidos na Tabela 9. Este valor é o ideal para guardar os *layers* diários na área geográfica (Portugal) e nos dados utilizados, o que pode variar de acordo com a localização em análise e com o número de fontes externas desejadas, valor este que é facilmente alterado na página de configuração do GeoWebCache.

### 5.3 Desempenho

Para o caso de um *layer* não estar pré-calculado, o tempo decorrido entre o momento em que é feito o pedido pode variar de acordo com a quantidade de informação e com o número de *layers* solicitados de uma só vez. No Gráfico 1, estão refletidos os tempos médios dos testes efetuados para o caso de se representar *layers* já pré-calculados ou fazer o seu cálculo e criação no momento em que



são pedidos. Para testar estas soluções foram solicitados: um único *layer* que representa o trânsito fluído; os três *layers*, que representam o trânsito fluído, pouco congestionado e congestionado; e um único *layer* que representa os valores de KPI das antenas. Os testes foram realizados usando um nível de *zoom* do mapa de 12 e este estava centrado na cidade do Porto. Para medir os tempos de cada *tile* foi utilizado a ferramenta de programador do *browser* Google Chrome (na secção “Rede”). Cada um dos testes foi executado 10 vezes e os valores mínimo, máximo e médio de cada pedido por ser analisado no Apêndice 8. No primeiro teste, que diz respeito ao trânsito fluído, os resultados obtidos foram semelhantes com ou sem *cache*, sendo viável ter uma solução sem *cache* se só estivesse a ser considerado este *layer*. Em seguida, utilizando os três *layers* do trânsito, os tempos de representação começam a ser superiores aos calculados no primeiro teste, no caso em que os *layers* já estão pré-calculados o tempo aumenta para o triplo do teste anterior, isto porque apesar dos *layers* serem pedidos todos de uma só vez pela interface, estes são representados de forma sequencial: em primeiro lugar o trânsito fluído, de seguida o trânsito pouco congestionado e por último o trânsito congestionado. No caso em que é necessário fazer criação dos *tiles*, o tempo de resposta aumenta, tornando a utilização uma pouco mais lenta. Por fim, no último teste a discrepância de valores com *cache* e sem *cache* é muito mais significativa. Uma das razões para que isto aconteça é o facto de haver muitas antenas localizadas na zona do Porto, o que faz com que os cálculos para a criação do *layer* aumentem expressivamente. Enquanto que com *cache* os valores voltam a descer para cerca de 0,5/0,6 segundos, sem *cache* os valores rondam os 26 segundos, o que torna a experiência de utilização muito pouco fluída.

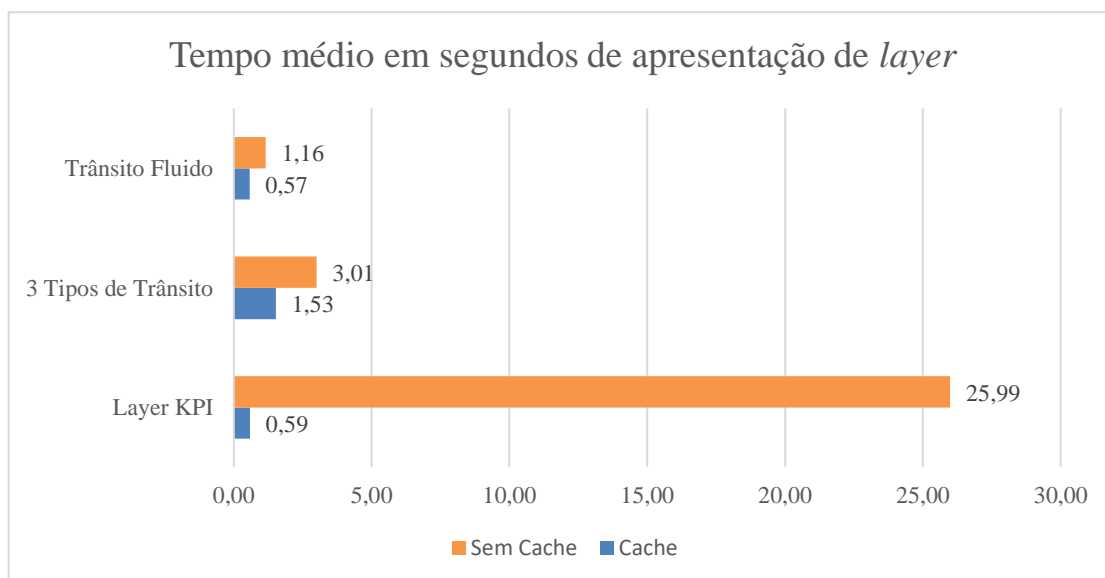


Gráfico 1 - Tempo médio em segundos de apresentação de *layers*

## 5.4 Discussão

Como analisado, o espaço de armazenamento ocupado para esta solução é bastante elevado quando são guardados níveis de *zoom* superiores a 15 e, estes valores agravam-se, quando o intervalo de tempo a guardar é superior a 24 horas. Este valor pode variar um pouco conforme o formato das imagens que definem os *tiles*, mas esta alteração não faz variar de forma drástica o espaço de armazenamento necessário para esta solução. No entanto, o espaço ocupado para guardar os dados na base de dados é consideravelmente sustentável a longo prazo.

Apesar dos *tiles* pré-calculados poderem ter um custo elevado no que diz respeito ao armazenamento necessário para o sistema, estes aquando da utilização da interface do cliente são uma mais valia no seu desempenho, tornando assim a sua utilização mais fluída e natural.

## 6 Conclusão e trabalho futuro

O presente trabalho, proposto pela Nokia Solutions and Networks (NSN), tinha como objetivo cruzar a informação da rede de telecomunicações com informação externa (meteorologia, trânsito, eventos), de forma a que fossem criadas novas formas e perspectivas de analisar a rede. A empresa, no seu portefólio, oferece a ferramenta Nokia Performance Manager, que permite fazer a análise histórica do estado da rede, querendo com o presente trabalho explorar a aquisição, manipulação e representação de informação georreferenciada de diferentes contextos.

Com este trabalho considera-se que a abordagem ao desafio proposto foi bem-sucedida, já que se alcançaram respostas aos problemas iniciais, sendo agora possível apresentar quatro conclusões:

- **Como foi obtida a informação.** Sendo o objetivo deste estágio de mestrado desenvolver e criar formas de extrair informação mais refinada e complexa, analisando e cruzando os dados da rede com outras fontes disponíveis, o primeiro desafio apresentado foi como obter esta informação. Relativamente à informação da rede foi necessário perceber como é obtida, tratada e de que forma deve ser armazenada para ser otimizada e disponibilizada. Depois de compreender como esta deveria ser tratada e armazenada foi necessário criar um simulador de valores para preencher os dados de rede, visto que, por questões de privacidade da empresa, não foi possível utilizar informação de clientes para testar qualquer tipo de solução. Numa segunda fase, como a empresa não tem acesso a informação que não diga respeito a redes de telecomunicações, foi necessário explorar os serviços *web* existentes para obter os dados externos que pudessem complementar os já existentes de forma a permitir uma análise mais detalhada do comportamento da rede. A informação externa considerada para esta fase foi o trânsito automóvel, a meteorologia e os eventos. Porém, numa fase final do trabalho apenas o trânsito automóvel e a meteorologia foram apresentados ao cliente. Para obter a informação das fontes externas foi criado um programa Java que com recurso a uma biblioteca externa faz o pedido da informação de forma a que o agendamento de processos seja feito de hora em hora para os contextos externos escolhidos.

- **Como foi tratada e armazenada a informação.** Uma vez reunidas todas as fontes de informação externas e os valores da rede de telecomunicações foi necessário perceber qual desta informação poderia vir a ser importante para a compreensão do comportamento da rede de telecomunicações, assim como, de que forma estes dados poderiam ser utilizados no resultado final. Sabendo que os dados da rede de telecomunicações já foram simulados de forma a que apenas os valores de interesse fossem criados, no caso das fontes externas foi necessário filtrar e tratar a informação obtida tendo sempre em conta a correlação dos dados, os casos de uso e a forma como serão disponibilizados e apresentados. Nesta fase foi necessário ter em atenção os dados que seriam tratados como dados geográficos e os que seriam como dados não geográficos. Para o armazenamento da informação da rede e das fontes externas foi escolhido o sistema de gestão de base de dados Oracle, que tem a capacidade de lidar com dados geográficos graças ao Oracle Spatial Data Option. Desta forma, os dados geográficos considerados como tal foram: a localização de cada antena de rede, as estradas que dizem respeito ao trânsito automóvel e os distritos que definem as zonas em análise na meteorologia.

- **Como foi disponibilizada a informação.** Esta fase o trabalho focou-se em como a informação seria organizada na base de dados para mais tarde ser disponibilizada para o analista. Foi necessário que a solução que disponibilizou os dados tivesse em conta que na base de dados existem dados geográficos e não geográficos e que estes necessitam de um tratamento diferenciado. Para disponibilizar a informação foi utilizado o Geoserver, um servidor de mapas que tem a capacidade de obter a informação geográfica da base de dados e disponibilizá-la utilizando *standards* estabelecidos pelo Open Geospatial Consortium para a troca de dados geográficos. Depois da instalação foi necessário configurar um *plugin* para ser possível a leitura de dados da base de dados Oracle. Foi escolhida a informação a ser apresentada e foi necessário criar estilos para que os dados geográficos fossem apresentados da forma pretendida, isto é, o trânsito como linhas de diferentes cores, as antenas como triângulos no mapa e a meteorologia com o ícone que define o estado tempo, centrado no respetivo distrito. O Geoserver, depois de todas as camadas criadas e prontas para serem disponibilizadas, só cria os *tiles* de cada uma no momento em que são solicitadas pelo cliente, o que em várias situações torna a utilização da interface demasiado lenta. Para tal, foi utilizado o Geowebcache, um servidor *cache* para *tiles*, para pré-calcular as camadas mais recentes de forma a que a disponibilização desta aquando um pedido é solicitado seja devolvida de forma quase imediata.

- **Como foi apresentada a informação.** Por fim, apesar da forma de apresentação da informação não ser o foco principal do trabalho, procurou-se uma forma ótima de apresentar toda a informação obtida tendo em conta as suas características e usabilidade. Um dos principais objetivos foi que toda a informação armazenada nas fases anteriores do trabalho pudesse ser apresentada. Esta interface final é caracterizada pelo mapa e por alguns filtros que permitem escolher a data e hora a

analisar e, conseqüentemente, a informação a ser apresentada no mapa, desde as antenas de rede e os seus indicadores-chave de desempenho (KPIs), aos contextos externos escolhidos. Na representação da informação, recorreu-se a cores ou ícones para representar um valor de um contexto, por esta razão é possível selecionar cada um destes elementos e ter acesso aos valores exatos do elemento, assim como alguma informação extra que possa vir a ser útil para a análise da rede.

Este estudo experimental tem o propósito de perceber se é viável apostar neste tipo de solução, isto é, cruzar os dados da rede de telecomunicações com dados de fontes externas, o que abre espaço a novas funcionalidades e melhorias para trabalho futuro até esta se tornar uma solução de nível empresarial. Um dos pontos que trará sempre valor a esta solução será a adição de novos contextos e novas fontes externas que possam vir a ter impacto na rede. Por exemplo, além dos contextos que já foram abordados neste relatório, situações como incêndios florestais, catástrofes naturais, estatísticas populacionais, etc. são sempre informações que podem ajudar a compreender o comportamento da rede. Porém, como a empresa não possui quaisquer fontes externas, estes serviços têm de existir no mercado, ser atualizados o mais regularmente possível e muitas vezes, quando estes existem, têm custos associados.

Outro alvo de melhoria que pode ser tido em conta para trabalho futuro são os pedidos solicitados ao Geoserver. Atualmente estes pedidos são definidos pelo nome do *layer*, nível de *zoom* e *bounding box* e o que acontece é que são enviados todos os *tiles* que pertencem à *bounding box* respetiva, independentemente do *tile* pedido ter informação ou não. Isto leva a que, num *layer* com muitos *tiles* transparentes, o tempo de resposta e a carga do Geoserver aumente sem que os recursos do mesmo estejam a ser gastos em informação útil. Assim, se no futuro fosse possível que o Geoserver ignorasse os *tiles* transparentes e apenas tratasse dos que contêm informação, poderia trazer uma grande melhoria de desempenho, assim como, no caso do Geowebcache, o espaço de armazenamento, que agora é gasto nos *tiles* transparentes, ficaria livre.

Como analisado nos testes e resultados do presente trabalho, o espaço ocupado para armazenar *tiles* é muito elevado, o que pode ser pouco sustentável para uma empresa quando se quer fazer um armazenamento dentro de um determinado intervalo de tempo. No entanto, observando com atenção as diferenças, de um nível de *zoom* do mapa para o nível de *zoom* seguinte ou anterior, estas não são significativas. O que leva a que, se nesta solução apenas se utilizasse os níveis de *zoom* do mapa de dois em dois (a título de exemplo: nível 1, 3, 5, 7, ...), o espaço de armazenamento ocupado diminuiria para cerca de metade.

Além disso, um dos problemas atuais é que quando um *layer* tem muitos pontos para serem representados no mapa, estes são representados tanto quando o nível de *zoom* é 5 como quando é 12, o que leva a que no primeiro caso, devido ao baixo nível de *zoom*, todos os pontos sejam apresentados

sobrepostos sem oferecer grande possibilidade de análise. No futuro uma solução para este problema seria criar um *cluster* de pontos quando o nível de zoom é inferior e ir dividindo este *cluster* em *clusters* mais pequenos conforme o *zoom* do mapa fosse sendo aumentado, o que tornaria a interface mais legível e menos confusa para o analista.

Por fim, com o trabalho desenvolvido até este ponto, a NSN pode completar e desenvolver novas funcionalidades ao protótipo, abrindo assim espaço a que esta nova forma de analisar a rede seja incluída no seu portefólio de ferramentas. A adição de novas funcionalidades como pesquisa de KPIs a serem apresentados por nome, novas formas de correlacionar os dados das fontes externas, seleccionar uma determinada área a ser analisada e criar de gráficos e estatísticas, seriam algumas das melhorias que poderiam ser adicionadas no trabalho futuro.

Todo este percurso consistiu numa desafiante experiência, que se traduziu num crescimento pessoal e profissional, uma vez que, dada a oportunidade de poder trabalhar num contexto empresarial, com profissionais com grande nível de experiência e conhecimento, e também pelo facto de perceber de que forma os métodos de trabalho e comunicação são feitos numa empresa tão bem-sucedida, foi uma mais valia para os desafios que se apresentarão no futuro. Além disto, ter a oportunidade de trabalhar num protótipo que pode vir a fazer parte do produto da empresa foi definitivamente uma benesse para o conhecimento pessoal e para o desenvolvimento de métodos de trabalho.

## 7 Referências

- [1] “Nokia in Portugal | Nokia Networks.” [Online]. Available: <https://networks.nokia.com/pt>. [Accessed: 13-Dec-2017].
- [2] Nokia Solutions and Networks, Product Description Nokia Solutions and Networks Performance Manager. Technical Report, 2014.
- [3] S. Chaudhuri and U. Dayal, “An overview of data warehousing and OLAP technology,” *ACM SIGMOD Rec.*, vol. 26, no. 1, pp. 65–74, 1997.
- [4] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. pp. 4, 2012.
- [5] S. Rivest, Y. Bédard, and P. Marchand, “Toward better support for spatial decision making: Defining the characteristics of spatial on-line analytical processing (SOLAP),” *Geomatica*, vol. 55, no. 4, pp. 539–555, 2001.
- [6] Franklin, C. “An Introduction to Geographic Information Systems: Linking Maps to Databases”. *Database*, April, pp. 13-21, 1992.
- [7] S. Rivest, Y. Bédard, M. J. Proulx, M. Nadeau, F. Hubert, and J. Pastor, “SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 60, no. 1, pp. 17–33, 2005.
- [8] M. MacEachren, “An evolving cognitive-semiotic approach to geographic visualization and knowledge construction,” *Inf. Des. J.*, vol. 10, no. Bertin 1983, pp. 26–36, 2001.
- [9] Bédard, Y. Spatial OLAP, Vidéoconférence. 2ème Forum annuel sur la R-D, Gématique VI: Un monde accessible, 13-14 novembre, 1997, Montréal.
- [10] Han, J., Stefanovic, N., Koperski, K. “Selective materialization: An Efficient Method for Spatial Data Cube Construction”. *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'98)*, Melbourne, Australia, 1998.
- [11] S. Rivest, Y. Bédard, M. J. Proulx, and M. Nadeau, “SOLAP: a New Type of User Interface To Support Spatio-Temporal Multidimensional Data Exploration and Analysis,” *Knowl. Creat. Diffus. Util.*, pp. 2–3, 2003.
- [12] J. Han, K. Koperski, and N. Stefanovic, “GeoMiner,” *ACM SIGMOD Rec.*, vol. 26, no. 2, pp. 553–556, 1997.

- [13] Shekhar, S., “Map Cube: A Visualization Tool for Spatial Data Warehouses”, Geographic Data Mining and Knowledge Discovery, 73-108. New York: Taylor & Francis, 2000.
- [14] P. H. Chris Stolte, Diane Tang, “Multiscale Visualization Using Data Cubes,” *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 2, pp. 176–187, 2003.
- [15] G. Colonese, R. Manhaes, and S. Gonzalez, “PostGeoOlap: An Open-Source Tool for Decision Support,” no. January, 2005.
- [16] M. Scotch and B. Parmanto, “SOVAT: Spatial OLAP visualization and analysis tool,” *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 0, no. C, p. 142-, 2005.
- [17] P. Compieta, S. Di Martino, M. Bertolotto, F. Ferrucci, and T. Kechadi, “Exploratory spatio-temporal data mining and visualization,” *J. Vis. Lang. Comput.*, vol. 18, no. 3, pp. 255–279, 2007.
- [18] S. Bimonte, A. Tchounikine, and M. Miquel, “Spatial OLAP: Open issues and a web based prototype,” *10th Agil. Int. Conf. ....*, no. figure 2, pp. 1–11, 2007.
- [19] R. Oliveira Lima da Silva, R. S. Manhães, and P. F. Gitahy, “WebGeoOlap : Visualização de dados multidimensionais e geográficos para ambiente web,” 2008.
- [20] N. Fidalgo, V. C. Times, J. Silva, F. De Souza, and A. C. Salgado, “Providing Multidimensional and Geographical Integration Based on a GDW and Metamodels ,” 2004.
- [21] V. R. T. Ferraz and M. T. P. Santos, “GlobeOLAP: Improving the geospatial realism in multidimensional analysis environment,” *ICEIS 2010 - Proc. 12th Int. Conf. Enterp. Inf. Syst.*, vol. 5 HCI, pp. 99–107, 2010.
- [22] S. Di Martino, S. Bimonte, M. Bertolotto, F. Ferrucci, and V. Leano, “Spatial online analytical processing of geographic data through the google earth interface,” *Stud. Comput. Intell.*, vol. 348, pp. 163–182, 2011.
- [23] K. A. Derbal, Z. Tahar, K. Boukhalifa, I. Frihi, and Z. Alimazighi, “From spatial data warehouse and decision-making tool to SOLAP generalisation approach for efficient road risk analysis,” *Int. J. Inf. Technol. Manag.*, vol. 15, no. 4, p. 364, 2016.
- [24] N. Gür, J. Nielsen, K. Hose, and T. B. Pedersen, “GeoSemOLAP: Geospatial OLAP on the Semantic Web Made Easy,” *WWW (Companion Vol.)*, pp. 213–217, 2017.
- [25] B. Evans and C. E. Sabel, “Open-Source web-based geographical information system for health exposure assessment,” *Int. J. Health Geogr.*, vol. 11, p. 2, 2012.
- [26] A. Scharl and K. (Klaus) Tochtermann, *The geospatial web : how geobrowsers, social software and the Web 2.0 are shaping the network society*. Springer, 2007.
- [27] J. de la Beaujardiere, “OpenGIS Web Map Server Implementation Specification,” *Organ. Environ.*, 2006.
- [28] P. E. Report, I. Simonis, and E. Report, “OpenGIS Web Feature Service 2.0 Interface Standard” *Engineering*, pp. 10–061, 2010.
- [29] P. Baumann, “OGC WCS 2.0 Interface Standard- Core: Corrigendum,” *OGC® Stand.*, pp. 1–57, 2012.
- [30] Anderson B, Deoliveira J: *WMS Performance Tests! Mapserver & Geoserver. Free and Open Source Software for Geospatial Victoria, Canada*; 2007
- [31] Jensen M: *Comparing apples and oranges: Uncovering the mystery of component selection in WebGIS*

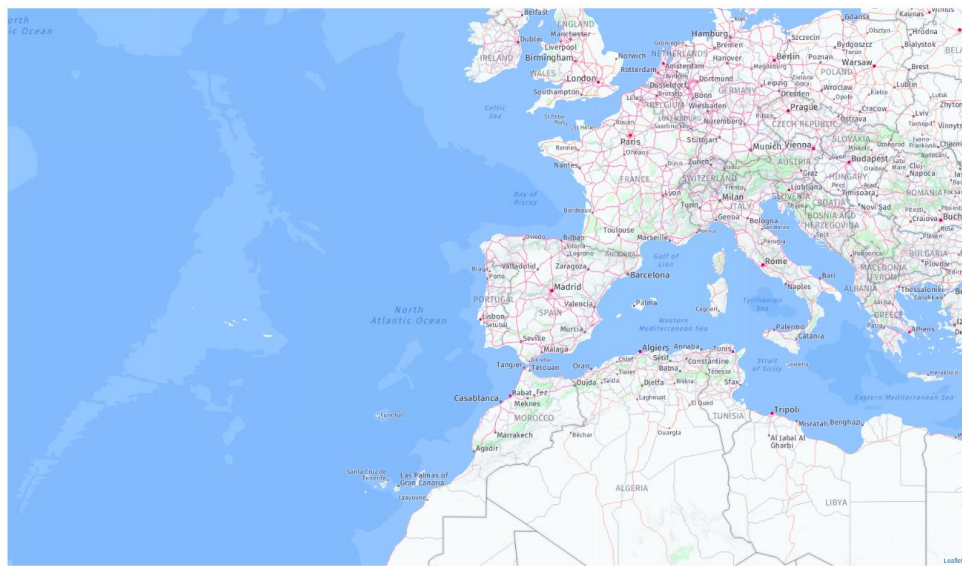


projects. Proceedings Free and Open Source Software for Geospatial Conference Sydney, Australia; 2009

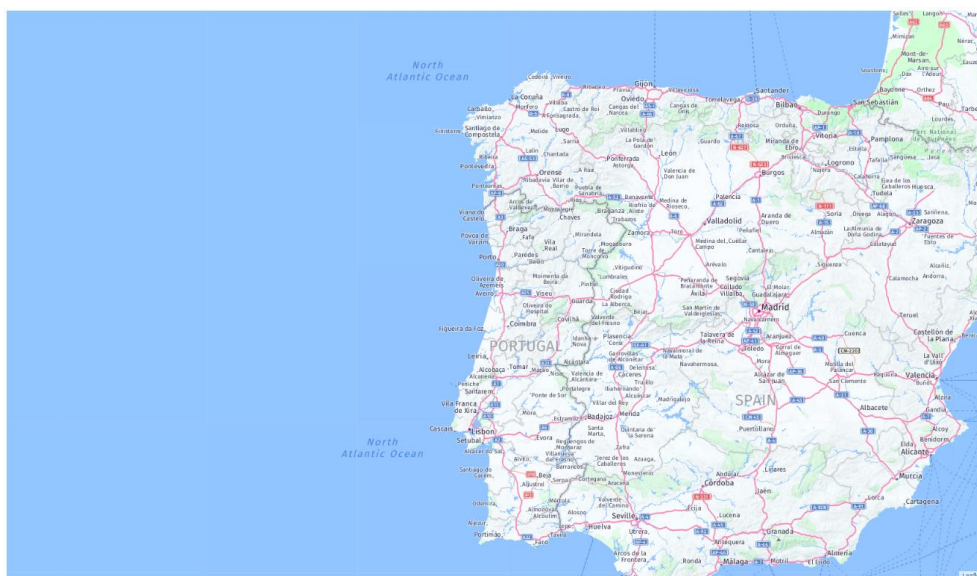
- [32] “GeoServer.” [Online]. Available: <http://geoserver.org/about/>. [Accessed: 10-Jan-2018].
- [33] “MapServer 7.0.7 documentation.” [Online]. Available: <http://mapserver.org/about.html>. [Accessed: 10-Jan-2018].
- [34] “AerisWeather.” [Online]. Available: <https://www.aerisweather.com/>. [Accessed: 13-Dec-2017].
- [35] “Price - OpenWeatherMap.” [Online]. Available: <http://openweathermap.org/price>. [Accessed: 13-Dec-2017].
- [36] “Dark Sky.” [Online]. Available: <https://darksky.net/dev>. [Accessed: 13-Dec-2017].
- [37] “Build apps with HERE Maps API and SDK Platform Access - HERE Developer.” [Online]. Available: <https://developer.here.com/>. [Accessed: 13-Dec-2017].
- [38] “Weather API | JSON Weather | World Weather Online.” [Online]. Available: <https://developer.worldweatheronline.com/>. [Accessed: 13-Dec-2017].
- [39] “Google Maps JavaScript API | Google Developers.” [Online]. Available: <https://developers.google.com/maps/documentation/javascript/>. [Accessed: 14-Dec-2017].
- [40] “TransLink Open API.” [Online]. Available: <https://developer.translink.ca/>. [Accessed: 14-Dec-2017].
- [41] “Walk Score API for web and mobile developers.” [Online]. Available: <https://www.walkscore.com/professional/api.php>. [Accessed: 14-Dec-2017].
- [42] “Events Feed, Concert & Event API - Eventful API.” [Online]. Available: <http://api.eventful.com/>. [Accessed: 14-Dec-2017].
- [43] “Bandsintown API — Bandsintown Manager.” [Online]. Available: <http://www.artists.bandsintown.com/bandsintown-api/>. [Accessed: 14-Dec-2017].
- [44] “Graph API.” [Online]. Available: <https://developers.facebook.com/docs/graph-api/>. [Accessed: 14-Dec-2017].
- [45] Oracle White Paper, “Spatial and Graph Analytics with Oracle Database 18c”, February 2018.
- [46] “GADM.” [Online]. Available: [https://gadm.org/download\\_country\\_v3.html](https://gadm.org/download_country_v3.html). [Accessed: 26-Abr-2018].
- [47] “Antenas Vodafone - Google Fusion Tables.” [Online]. Available: [https://fusiontables.google.com/DataSource?docid=1P30XTsIJyCeU9fSuRdl\\_XSoCaM7OFNt6H-KeJe8#map:id=3](https://fusiontables.google.com/DataSource?docid=1P30XTsIJyCeU9fSuRdl_XSoCaM7OFNt6H-KeJe8#map:id=3). [Accessed: 06-Jun-2018].
- [48] “Quartz Enterprise Job Scheduler.” [Online]. Available: <http://www.quartz-scheduler.org/>. [Accessed: 06-Jun-2018].
- [49] “Introduction to SLD — GeoServer 2.13.x User Manual.” [Online]. Available: <http://docs.geoserver.org/stable/en/user/styling/sld/introduction.html>. [Accessed: 06-Jun-2018].
- [50] “Using WMS and TMS services - Leaflet - a JavaScript library for interactive maps.” [Online]. Available: <https://leafletjs.com/examples/wms/wms.html>. [Accessed: 06-Jun-2018].



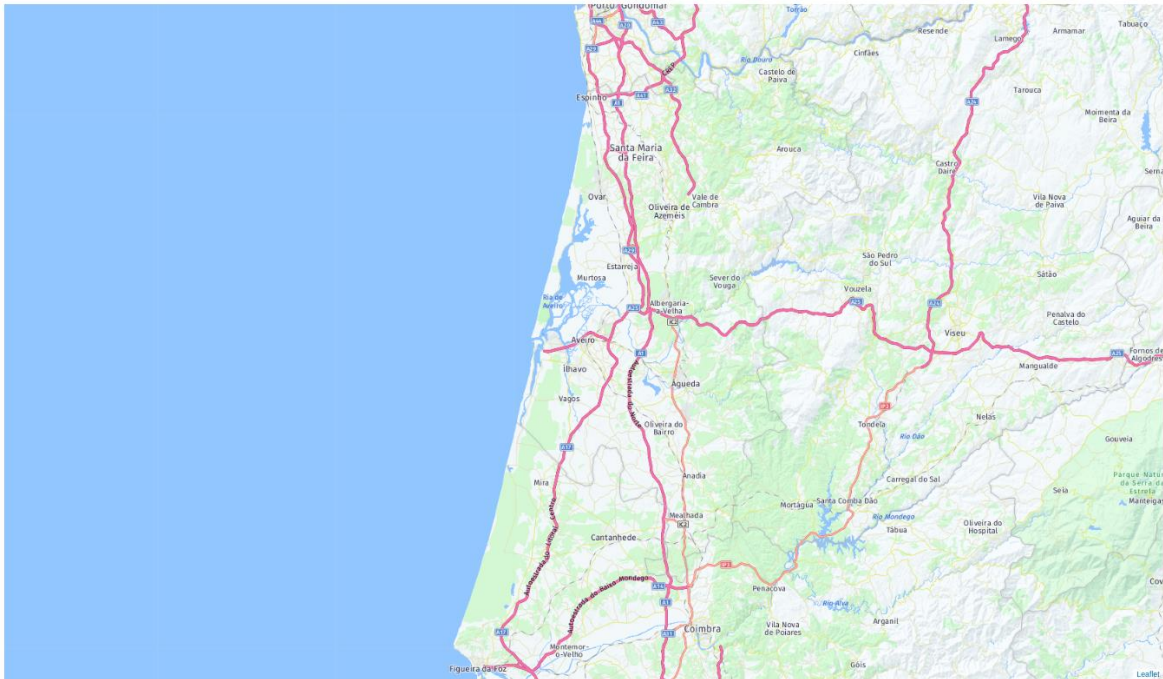
## 8 Apêndices



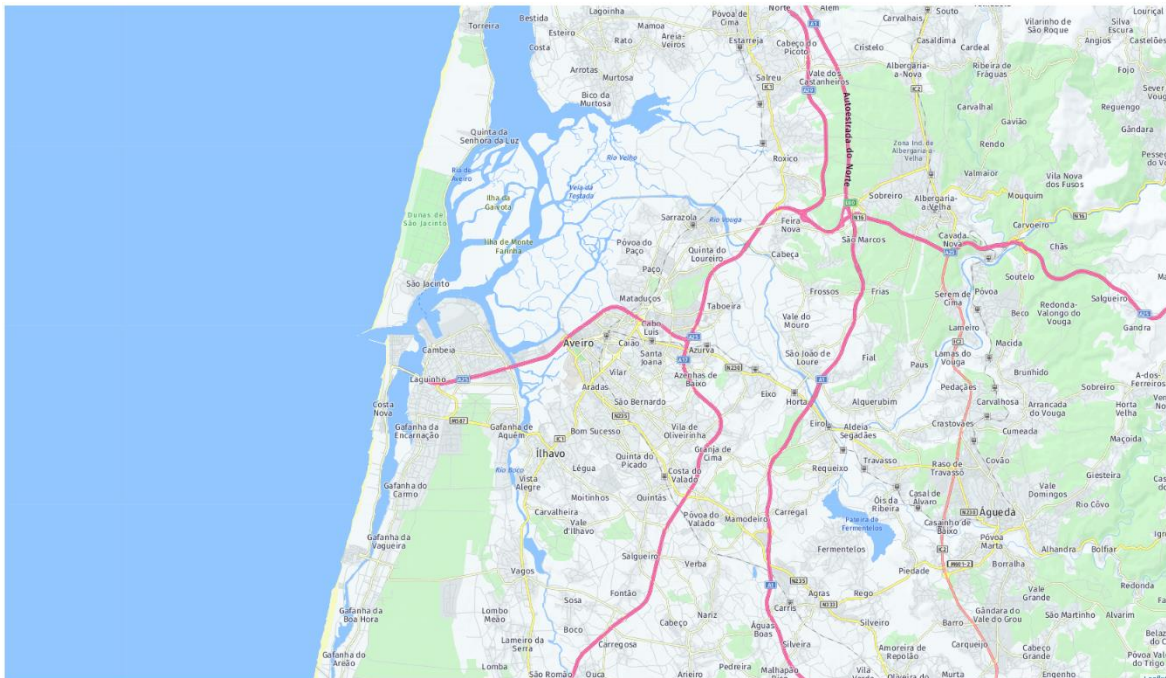
Apêndice 1 - Mapa com o nível de zoom 5



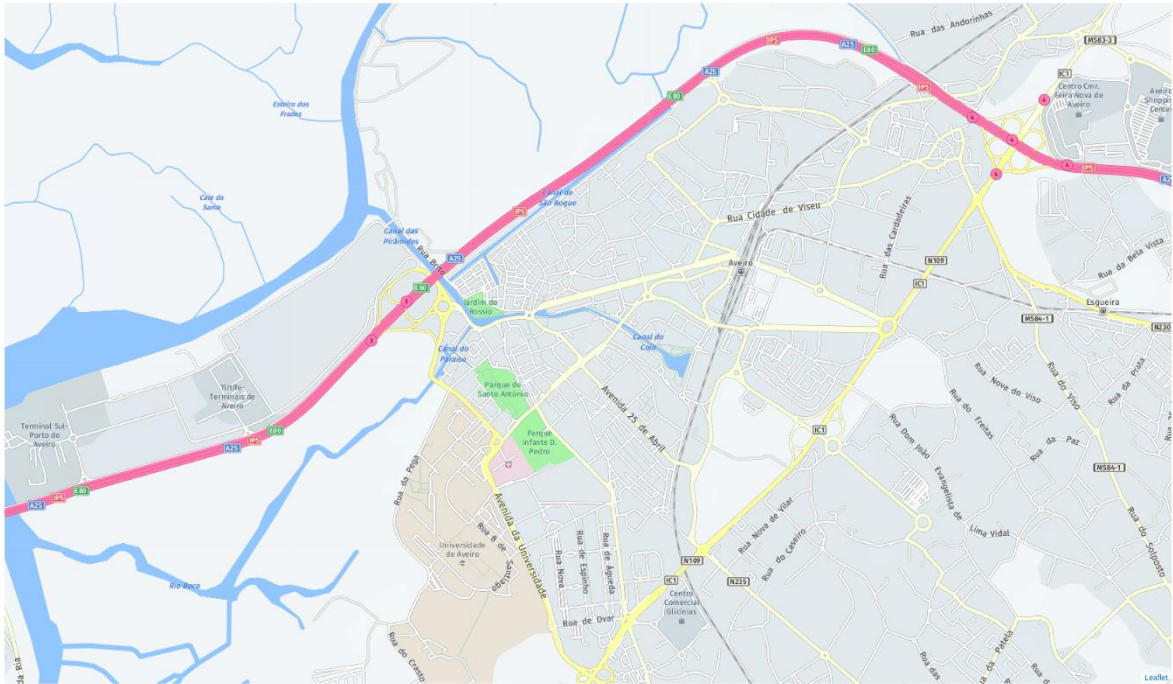
Apêndice 2 - Mapa com o nível de zoom 7



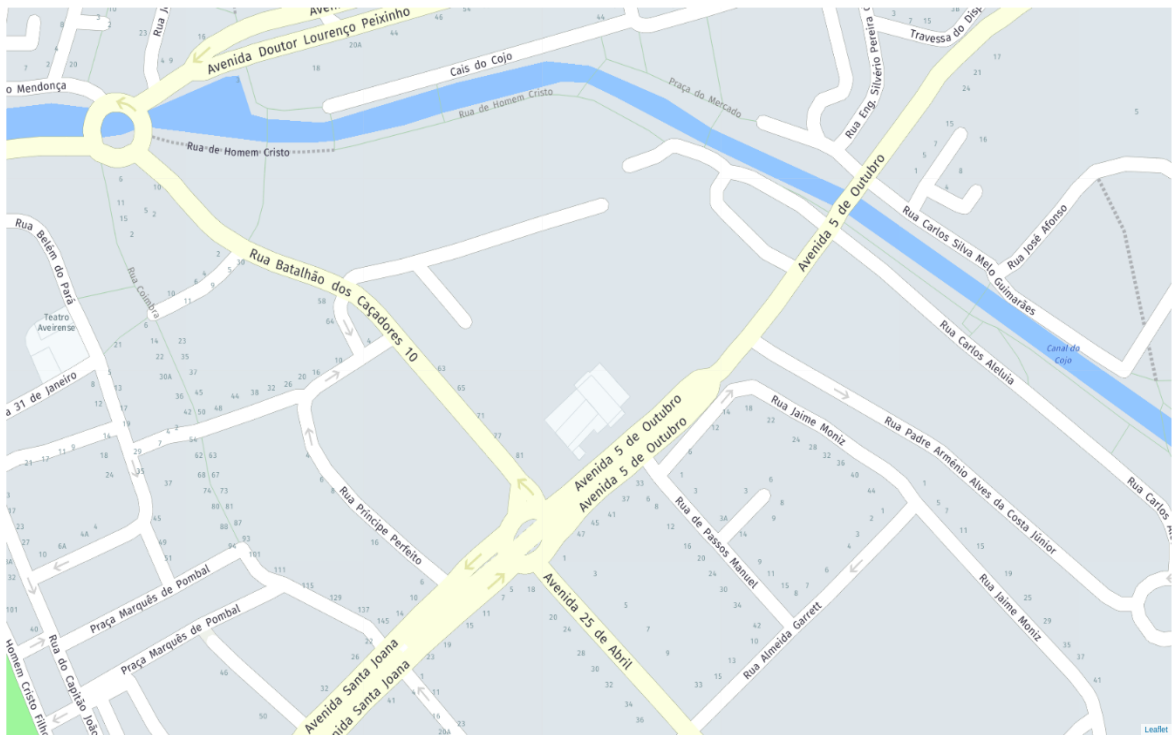
Apêndice 3 - Mapa com o nível de zoom 10



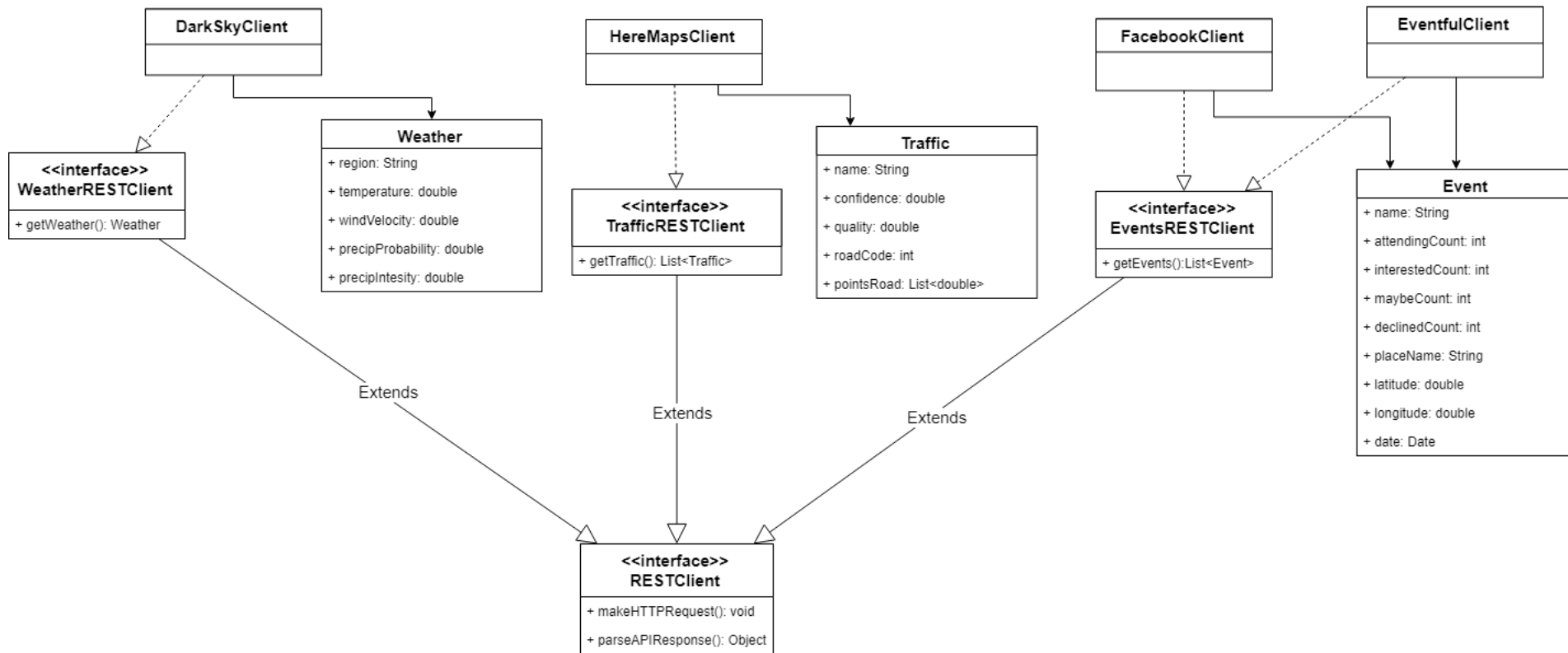
Apêndice 4 - Mapa com o nível de zoom 12



Apêndice 5 - Mapa com o nível de zoom 15



Apêndice 6 - Mapa com o nível de zoom 21



Apêndice 7 - Diagrama de classes da aplicação das APIs externas

Apêndice 8 - Resultados detalhados dos testes de desempenho

	Nº Teste	Cache			Sem Cache		
		Mínimo	Médio	Máximo	Mínimo	Médio	Máximo
<b>Trânsito fluído</b>	1	0,355	0,52	0,977	0,769	1,12	1,44
	2	0,299	0,62	0,939	0,801	1,1	1,5
	3	0,281	0,58	0,912	0,716	1,22	1,31
	4	0,316	0,51	1,01	0,759	1,18	1,47
	5	0,359	0,601	1,17	0,699	1,09	1,38
	6	0,304	0,59	0,886	0,723	1,15	1,41
	7	0,283	0,579	0,89	0,789	1,13	1,52
	8	0,389	0,582	0,992	0,694	1,11	1,35
	9	0,293	0,593	0,878	0,724	1,4	1,61
	10	0,294	0,558	0,904	0,756	1,11	1,41
<b>3 Tipos de Trânsito</b>	1	0,419	1,49	2,48	0,743	3,04	5,64
	2	0,377	1,52	2,35	0,812	2,96	4,9
	3	0,39	1,57	2,41	0,767	2,99	5,12
	4	3,7	1,48	2,36	0,787	3,1	5,53
	5	0,466	1,49	2,34	0,801	3,02	5,55
	6	0,346	1,67	3,36	0,832	3,12	5,61
	7	0,357	1,56	2,38	0,759	3,01	5,71
	8	0,4	1,47	2,37	0,774	2,94	5,01
	9	0,412	1,53	2,32	0,798	2,97	5,24
	10	0,377	1,49	2,58	0,82	2,91	5,33
<b>Layer KPI</b>	1	0,304	0,565	0,911	13,08	25,7	52,7
	2	0,382	0,537	0,962	13,03	26,1	53,29
	3	0,284	0,557	0,895	13,01	26,2	52,85
	4	0,286	0,602	0,949	12,76	25,8	53,21
	5	0,294	0,623	0,943	13,12	26,4	54,21
	6	0,307	0,558	0,951	13,15	26,5	54,71
	7	0,268	0,678	1,01	12,98	25,6	53,36
	8	0,38	0,582	0,928	12,94	25,8	53,36
	9	0,298	0,563	0,892	13,05	25,7	53,42
	10	0,388	0,64	1,01	13,13	26,1	54,01