



Sistemas Autónomos

Trabalho Prático 2

Grupo 1

16 de maio de 2021



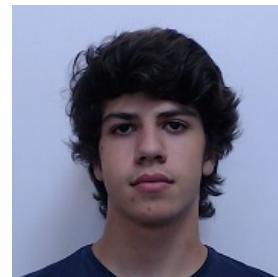
Ana Gil
A85266



Beatriz Rocha
A84003



Hugo Matias
A85370



João Abreu
A84802

Conteúdo

1	Introdução	4
2	Recolha e Armazenamento dos Dados	5
3	Tratamento dos Dados	8
4	Modelos de <i>Machine Learning</i>	9
5	Visualização dos Dados	13
6	Aplicação Web	16
6.1	<i>Home</i>	16
6.2	<i>Data Sources</i>	16
6.3	<i>Incidents</i>	17
6.4	<i>Events</i>	18
6.5	<i>Coordinates</i>	18
6.6	<i>Data Visualization</i>	18
6.7	<i>Geovisualization</i>	19
6.8	<i>Graphs and Charts</i>	20
6.9	<i>Predictions</i>	20
6.10	Menu de pesquisa	21
7	Conclusão	22

Lista de Figuras

1.1	Arquitetura do sistema desenvolvido	4
2.1	Processo de recolha dos dados	5
4.1	Atraso de acordo com o tamanho do incidente	10
4.2	Número de incidentes de acordo com as datas	10
4.3	Número de incidentes de acordo com a origem	11
4.4	Número de incidentes de acordo com o destino	11
4.5	Número de incidentes para as horas de uma data específica	12
4.6	Número de incidentes para as últimas 24 horas	12
5.1	Página 1 - Geovisualização	13
5.2	Página 2 - Gráficos	14
5.3	Página 3 - Previsões	15
6.1	<i>Home</i>	16
6.2	<i>Data Sources</i>	17
6.3	<i>Incidents</i>	17
6.4	<i>Events</i>	18
6.5	<i>Coordinates</i>	18
6.6	<i>Data Visualization</i>	19
6.7	<i>Geovisualization</i>	19
6.8	<i>Graphs and Charts</i>	20
6.9	<i>Predictions</i>	21
6.10	Menu de pesquisa	21

Lista de Tabelas

3.1 Ficheiro <code>coordinates.csv</code>	8
3.2 Ficheiro <code>events.csv</code>	8
3.3 Ficheiro <code>incidents.csv</code>	8

1 Introdução

Com o aumento exponencial da densidade populacional nos grandes centros urbanos, são necessárias soluções inteligentes que permitam gerir e resolver, de forma eficiente e eficaz, todos os problemas subjacentes a estas cidades. Nesse sentido, a segurança rodoviária tornou-se num crescente ponto de preocupação da nossa sociedade, facilmente explicado pelo número substancial de acidentes e fatalidades que se verificam nas estradas, dia após dia.

Uma *Smart City* deve ser capaz de responder a vários problemas, incluindo a segurança rodoviária, utilizando, para isso, sensores para coleção de dados, aplicando técnicas de fusão sensorial e fazendo uso de modelos de *Machine Learning* capazes de prever fenómenos futuros.

Assim sendo, este trabalho prático tem como objetivo a construção de uma plataforma que se pretende transversal a todos estes pontos:

- Conceber colecionadores de dados aplicados a sensores virtuais, focando cidades portuguesas;
- Explorar e tratar os dados, aplicando, entre outras, técnicas de geovisualização para análise dos dados geoespaciais coletados;
- Conceber e implementar modelos de *Machine Learning* para previsão.

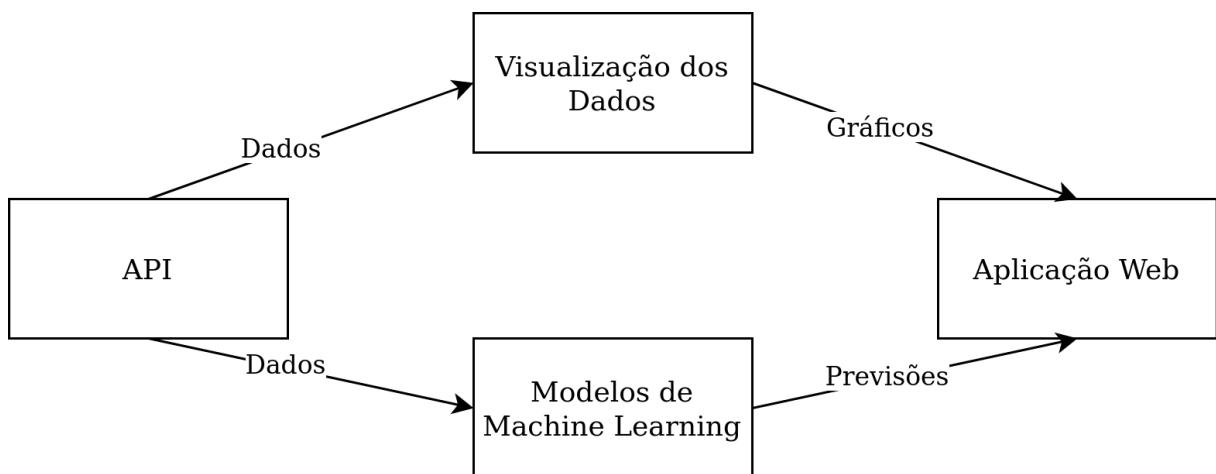


Figura 1.1: Arquitetura do sistema desenvolvido

2 Recolha e Armazenamento dos Dados

Tendo em conta que o nosso grupo se trata de um grupo de número ímpar, ficámos responsáveis por trabalhar dados relativos a incidentes rodoviários.

Devido à falta de sensores físicos, este processo é efetuado de forma virtual, com recurso a uma API disponibilizada pela TomTom que pode ser consultada em <https://developer.tomtom.com/traffic-api/traffic-api-documentation-traffic-incidents/incident-details>.

O processo de recolha dos dados é feito através de um *script* desenvolvido na linguagem de programação Python (esta escolha deve-se ao facto de a mesma proporcionar uma rápida prototipagem graças à sua simplicidade e vasta oferta de bibliotecas e ser bastante intuitiva quando a tarefa consiste em criar *scripts*) e pode ser resumido, essencialmente, em quatro etapas:

1. Fazer um pedido à API com todos os campos disponíveis (nesta etapa, é importante referir que, na tentativa de colecionar uma grande quantidade de incidentes, a sensorização incide sobre a zona de Lisboa, logo, no pedido, são utilizadas as coordenadas (38.812612, -9.263105) e (38.591305, -9.035139) que, por sua vez, compreendem a mesma);
2. Armazenar o pedido na base de dados (neste caso, na base de dados documental MongoDB, uma vez que as respostas aos pedidos feitos à API são em formato JSON e esta permite mapeamento direto entre os objetos JSON e os documentos);
3. Esperar 1 minuto (visto que o limite de pedidos diários à API é 2500);
4. Voltar à etapa 1.

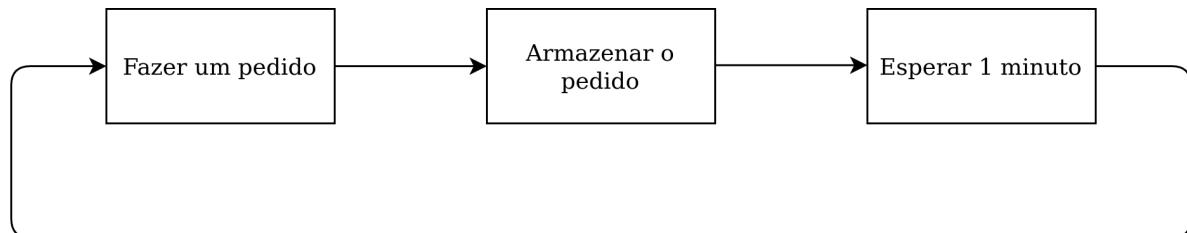


Figura 2.1: Processo de recolha dos dados

De seguida, é possível observar um exemplo de uma resposta a um pedido feito à API.

```
1 {  
2   "_id"::  
3     "$oid": "60854a2badbe3a63a8cea488"  
4   },  
5   "type": "Feature",  
6   "properties"::  
7     "id": "a5c3f205e271d2b3abd57cd872376756",  
8     "iconCategory"::  
9       "$numberInt": "8"  
10    },  
11    "magnitudeOfDelay"::  
12      "$numberInt": "4"  
13    },  
14    "startTime": "2021-04-25T07:00:00Z",  
15    "endTime": "2021-04-25T19:00:00Z",  
16    "from": "Ajuda",  
17    "to": "Calcadado Galvao",  
18    "length":  
19  }  
20
```

```

19         "$numberDouble": "113.2140078202"
20     },
21     "delay": {
22       "$numberInt": "0"
23     },
24     "roadNumbers": [
25
26   ],
27     "events": [
28       {
29         "code": {
30           "$numberInt": "401"
31         },
32         "description": "Closed",
33         "iconCategory": {
34           "$numberInt": "8"
35         }
36       },
37       {
38         "code": {
39           "$numberInt": "701"
40         },
41         "description": "Roadworks",
42         "iconCategory": {
43           "$numberInt": "9"
44         }
45       }
46     ],
47     "aci": null
48   },
49   "geometry": {
50     "type": "LineString",
51     "coordinates": [
52       [
53         [
54           {
55             "$numberDouble": "-9.2010592365"
56           },
57           {
58             "$numberDouble": "38.7044692228"
59           }
60         ],
61         [
62           {
63             "$numberDouble": "-9.2019121789"
64           },
65           {
66             "$numberDouble": "38.7045322788"
67           }
68         ],
69         [
70           {
71             "$numberDouble": "-9.2023601078"
72           },
73           {
74             "$numberDouble": "38.7045335871"
75           }
76         ]
77       ]
78     }

```

Posto isto, é importante referir que, de modo a evitar o armazenamento de objetos repetidos, antes de inserir um novo documento na base de dados, verificamos sempre se já lá existe algum com o mesmo identificador (que pode ser consultado no campo `properties`).

Por último, só nos resta mencionar que, de modo a recolher uma grande quantidade de dados e visto que isto é útil não só para podermos tirar conclusões acerca dos mesmos, mas também para que os nossos modelos de *Machine Learning* possam ter um bom desempenho, optámos por executar o *script* num servidor durante cerca de duas semanas.

3 Tratamento dos Dados

Visto que, mais à frente, iremos recorrer à plataforma Google Data Studio e esta não possui nenhum conector para MongoDB, a solução em que pensámos para contornar o problema consistiu em transformar o modelo de dados documental em relacional, para que fosse possível importar os dados através de ficheiros CSV.

Assim sendo, primeiro, começámos por eliminar o campo `roadNumbers`, visto que este possui informação redundante. Depois, visto que os campos `coordinates` e `events` se tratam de *arrays*, deram origem a dois ficheiros CSV (`coordinates.csv` e `events.csv`, respetivamente) cujas entradas se referem ao respetivo objeto através do seu identificador, ou seja, para cada *array* do *array* de coordenadas formado pela longitude e pela latitude, foi criada uma entrada com esses mesmos atributos que faz referência ao respetivo objeto e, para cada elemento do *array* de eventos formado pelo código, pela descrição e pela categoria, de forma análoga, foi criada uma entrada com esses mesmos atributos que faz referência ao respetivo objeto. Quanto aos restantes atributos, foi possível colocá-los diretamente num único ficheiro CSV (`incidents.csv`).

Tabela 3.1: Ficheiro `coordinates.csv`

Atributo	Tipo
incident_id	String
lon	Float
lat	Float

Tabela 3.2: Ficheiro `events.csv`

Atributo	Tipo
incident_id	String
event_code	Integer
event_description	String
event_iconCategory	Integer

Tabela 3.3: Ficheiro `incidents.csv`

Atributo	Tipo
_id	String
type	String
properties_id	String
properties_iconCategory	Integer
properties_magnitudeOfDelay	Integer
properties_startTime	Datetime
properties_endTime	Datetime
properties_from	String
properties_to	String
properties_length	Float
properties_delay	Integer
geometry_type	String

4 Modelos de *Machine Learning*

De modo a realizar previsões e tendo em conta a natureza quantitativa dos dados a prever, surgiu a ideia de aplicar o modelo KNN Regressor, de modo a resolver estes problemas de regressão.

Para efeitos de treino, foi necessário processar e filtrar o *dataframe* correspondente ao ficheiro `incidents.csv`. Esta filtração corresponde à eliminação das colunas `_id`, `type`, `properties_id`, `properties_magnitudeOfDelay`, `properties_endTime`, `properties_aci` e `geometry_type` e à eliminação de linhas com valores nulos.

Foram também criadas funções de auxílio que foram aplicadas de forma diferente dependendo das *features* e *target variable* usadas. Estas funções são as seguintes:

- `new_train_test_sets()`: Cria dados de treino e de teste aleatoriamente;
- `encodeData(train_data,test_data)`: Codifica os *datasets* de treino e de teste;
- `normalizeData(train_data,test_data)`: Normaliza os *datasets* de treino e de teste;
- `knn_regressor(datasets)`: Aplica o KNN Regressor para obter previsões;
- `print_series_prediction(y_test,predic,title,label1,label2)`: Produz um gráfico tendo em conta o resultado da função `knn_regressor(datasets)`.

Para além das funções apresentadas em cima, foram também criadas funções que modificam e alteram os *datasets* de treino e de teste, de modo a obter o *dataset* necessário com as *features* e *target variable* pretendidas:

- `data_length_delay()`: Prevê o atraso causado por um incidente tendo em conta o tamanho do incidente em metros;
- `data_startTime()`: Prevê o número de incidentes tendo em conta uma data;
- `data_from()`: Prevê o número de incidentes tendo em conta a origem;
- `data_to()`: Prevê o número de incidentes tendo em conta o destino;
- `searchData(spec_date)`: Prevê o número de incidentes numa hora de um dia específico;
- `last_24(data)`: Prevê o número de incidentes nas últimas 24 horas.

Aplicando o modelo KNN Regressor aos *datasets* acima apresentados, obtém-se os seguintes resultados com as métricas MAE (*Median Absolute Error*) e EVS (*Explained Variance Score*):

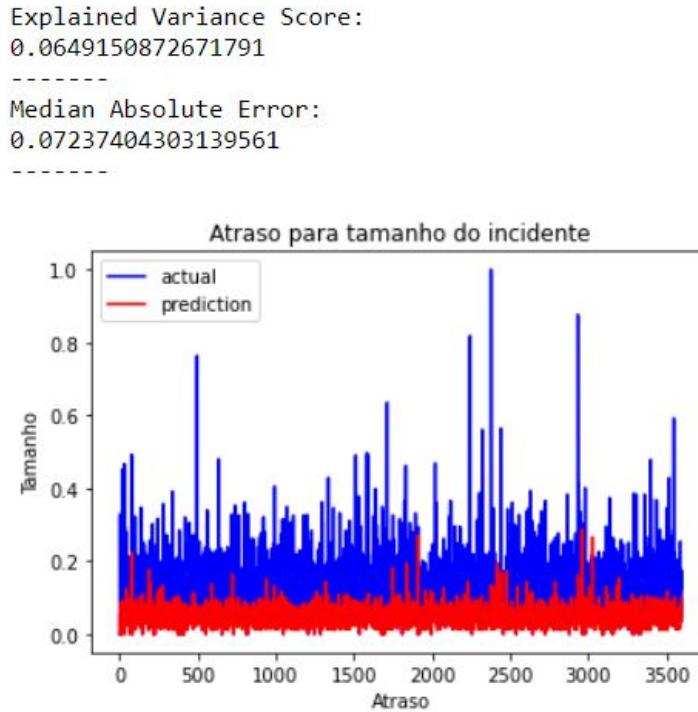


Figura 4.1: Atraso de acordo com o tamanho do incidente

Para a previsão do atraso tendo em conta o tamanho do incidente, o modelo KNN não é uma boa escolha para os dados em questão, pois, apesar de ter um MAE baixo, o seu EVS é também muito baixo, o que indica uma baixa força de associação entre o modelo e os dados.

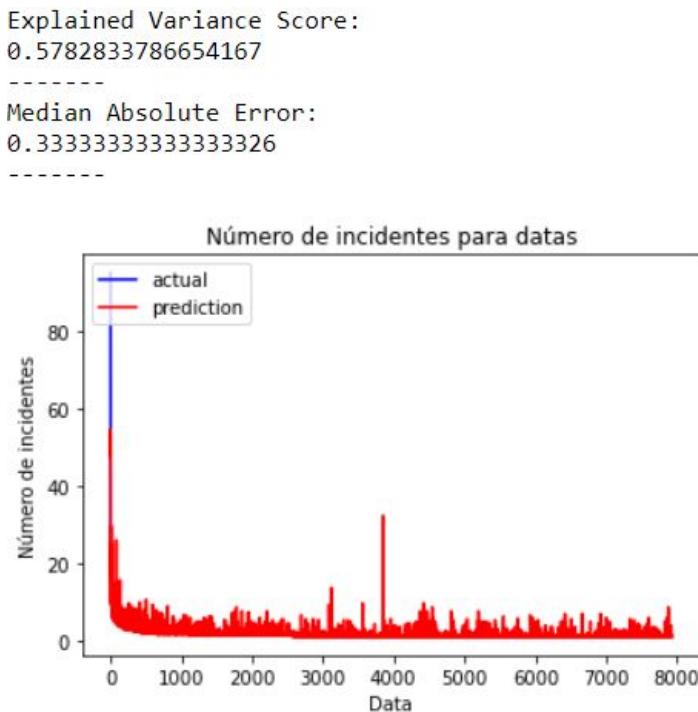


Figura 4.2: Número de incidentes de acordo com as datas

Com um EVS perto de 60% e MAE baixo é possível concluir que o modelo KNN é uma escolha razoável para realizar previsões com o *dataset* em questão.

Explained Variance Score:

0.3861822005781457

Median Absolute Error:

1.3333333333333335

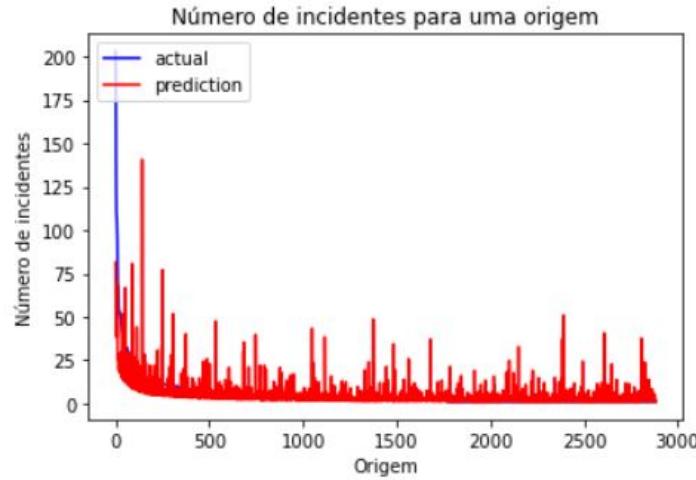


Figura 4.3: Número de incidentes de acordo com a origem

Com um EVS perto de 40% e MAE alto é possível concluir que o modelo KNN não é uma boa escolha para realizar previsões com o *dataset* em questão.

Explained Variance Score:

0.5123538902847345

Median Absolute Error:

1.6666666666666665

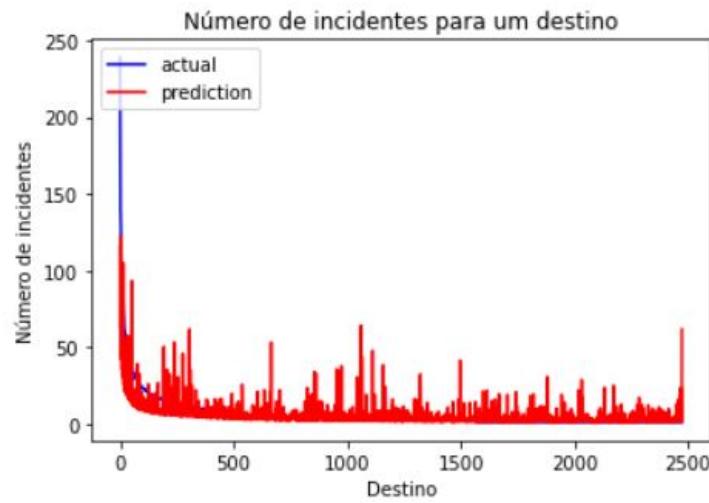


Figura 4.4: Número de incidentes de acordo com o destino

Com um EVS perto de 50% e MAE alto é possível concluir que o modelo KNN não é uma boa escolha para realizar previsões com o *dataset* em questão.

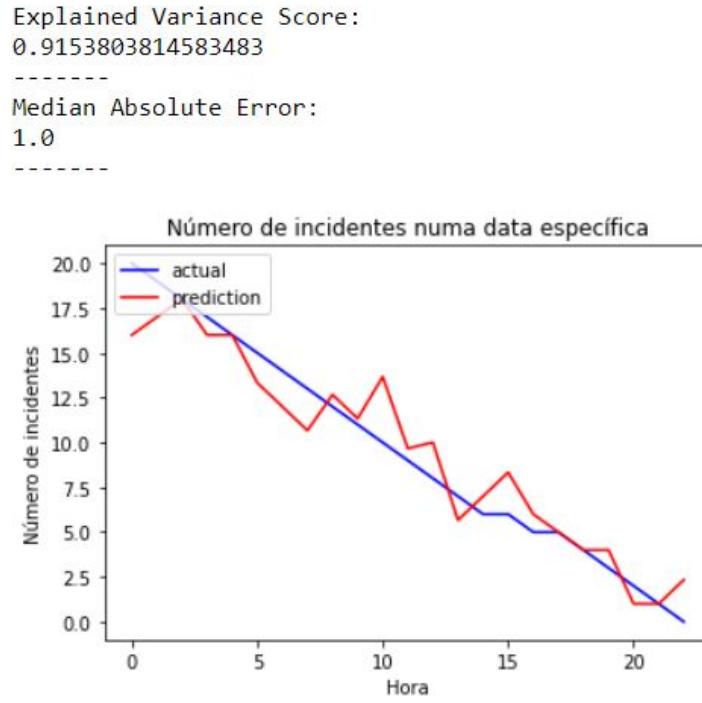


Figura 4.5: Número de incidentes para as horas de uma data específica

Apesar do MAE ser relativamente alto, o EVS perto de 92% demonstra que o modelo se adaptou extremamente bem a este *dataset*. A data em questão é 2021-04-26.

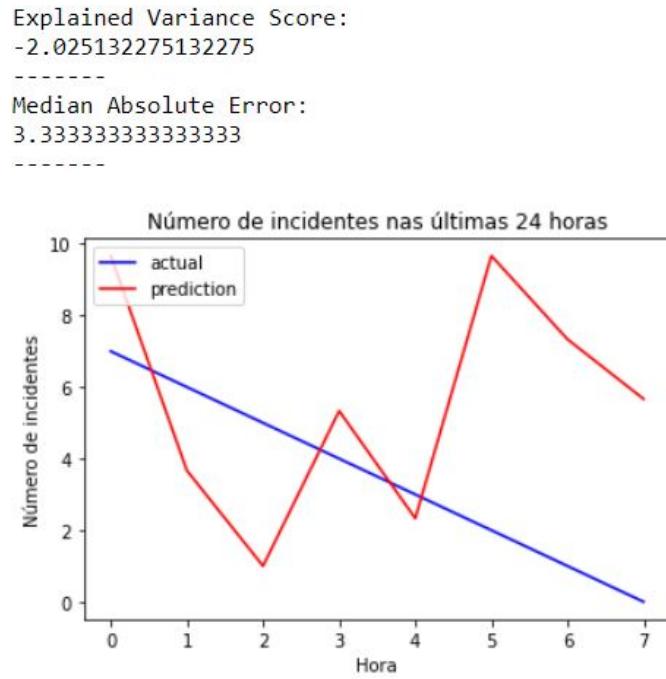


Figura 4.6: Número de incidentes para as últimas 24 horas

Com um MAE extremamente alto e um EVS negativo é possível concluir que o modelo KNN não é a melhor opção para previsões com este *dataset*.

5 Visualização dos Dados

Nesta secção, iremos explicar todo o processo desenvolvido no que toca à visualização dos dados e explicar brevemente as conclusões que se podem tirar de certos gráficos.

Desde já, é importante mencionar que, para isso, recorremos à plataforma Google Data Studio, uma vez que esta não apresenta custos associados, permite que várias pessoas editem os mesmos gráficos (desde que tenham permissões para tal) e revela-se bastante intuitiva, na medida em que consiste num processo *drag and drop*.

Para além disso, de modo a conseguirmos representar o número de incidentes por hora, convém referir que foi necessário realizar mais uma pequena manipulação dos dados. Do nosso ponto de vista, um incidente que tenha decorrido de x até y horas, significa que ocupou todas as horas entre x e y , inclusive, *i.e.*, um incidente que tenha começado às 11:00h e tenha durado até às 13:00h, aconteceu às 11:00h, às 12:00h e às 13:00h. Assim sendo, criámos um novo ficheiro CSV (`incidents_per_hour.csv`), onde, para cada incidente, colocámos as horas em que o mesmo aconteceu.

Posto isto, decidimos criar três páginas distintas: uma só com elementos de geovisualização, dando, assim, ênfase a esta componente, outra que exibe os dados através de vários elementos de análise estatística, como tabelas, gráficos de linhas, gráficos circulares, *etc.* e uma última com dados relativos a previsões, apresentados na secção 4.

De salientar que, para as duas primeiras páginas, no canto superior direito, encontra-se um elemento que representa um controlo de intervalo de datas, ou seja, é possível escolher as datas de início e de fim, de modo a filtrar os dados a analisar por data.

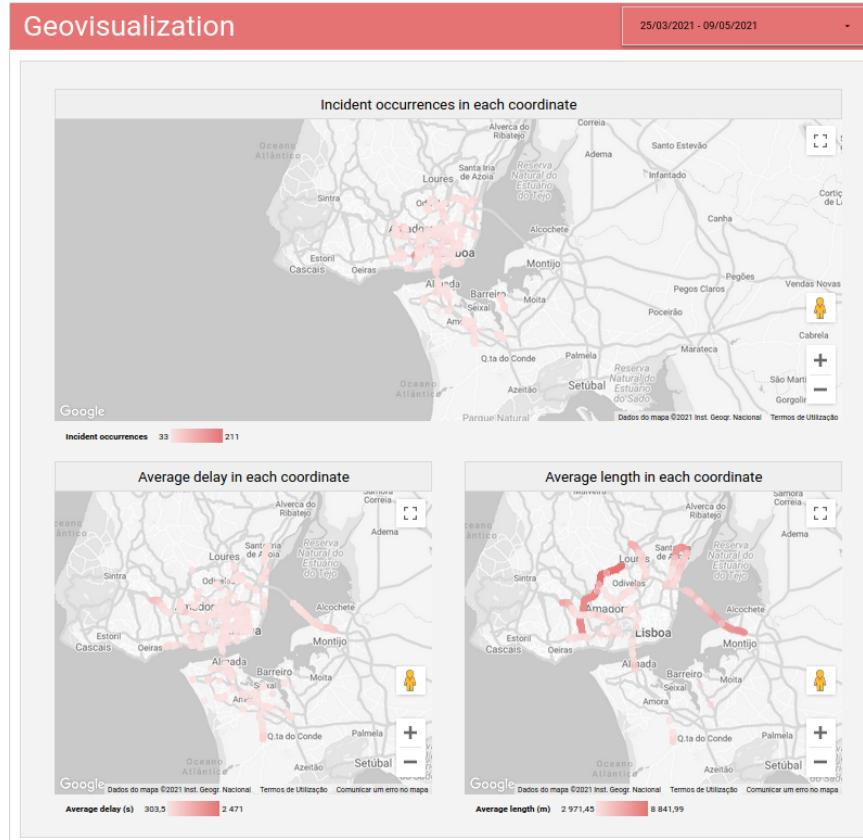


Figura 5.1: Página 1 - Geovisualização

Na imagem acima, encontra-se a primeira página desenvolvida que, por sua vez, diz respeito aos elementos de geovisualização. São apresentados três tipos de mapas utilizando gráficos de dispersão. Para cada um deles é feita uma combinação de dados entre os ficheiros `incidents.csv` e `coordinates.csv`, uma vez que um disponibiliza a variável em análise e o outro disponibiliza as coordenadas a representar no mapa, sendo estas interligadas através do identificador do incidente presente em ambos os ficheiros. O gráfico que se encontra no topo apresenta o número de incidentes para cada coordenada, pois diferentes incidentes podem conter coordenadas em comum. De seguida, optámos por usar a mesma lógica para desenvolver os dois gráficos de baixo que, para cada coordenada, têm em conta todos os valores de `delay` (tempo de atraso que o incidente provocou) e de `length` (comprimento do incidente), respetivamente, e calculam o seu valor médio.

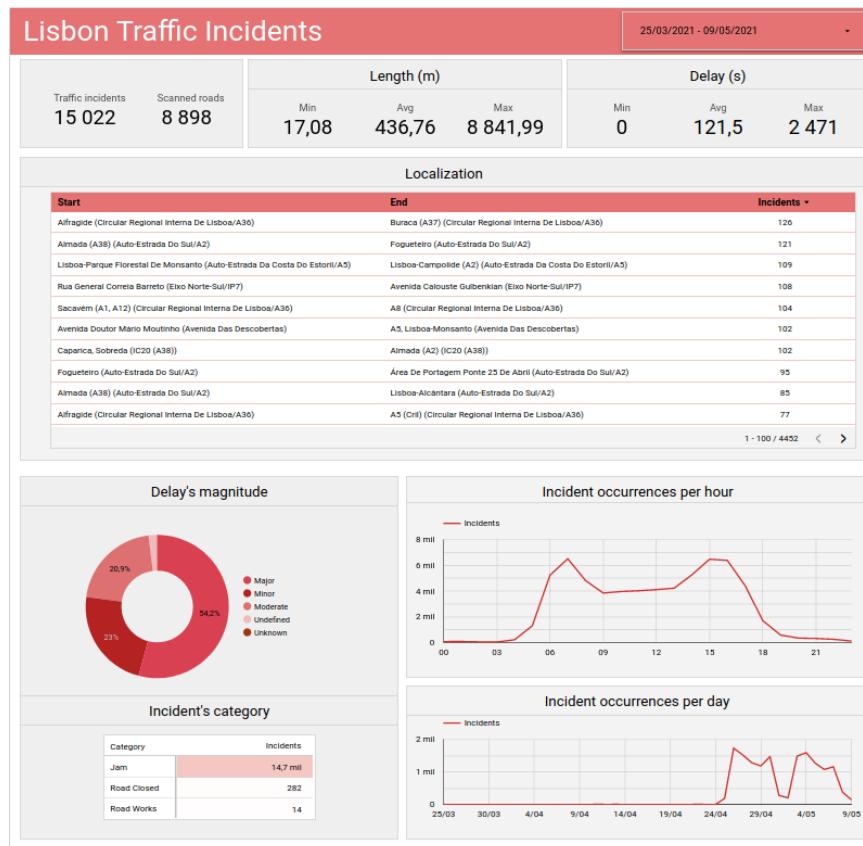


Figura 5.2: Página 2 - Gráficos

Na imagem acima, está presente a nossa segunda página de visualização dos dados.

No topo, encontra-se um conjunto de estatísticas gerais sobre os incidentes, nomeadamente o seu número total e o número de ruas onde os mesmos aconteceram. Depois, tanto para o atributo `length` como para o atributo `delay`, apresentamos o respetivo valor mínimo, médio e máximo. Na tabela imediatamente abaixo, temos o número de incidentes com início numa rua e fim noutra, com o intuito de apresentar alguma visualização dos dados no espaço. Em baixo, na coluna da esquerda, para cada magnitude do atributo `delay`, temos a quantidade de incidentes que aconteceram, representados sob a forma de percentagem e, na tabela de baixo, o número de incidentes para cada motivo do mesmo (`iconCategory`). Por fim, na coluna da direita, conseguimos ver a oscilação de ocorrência de incidentes ao longo das horas e ao longo dos dias.

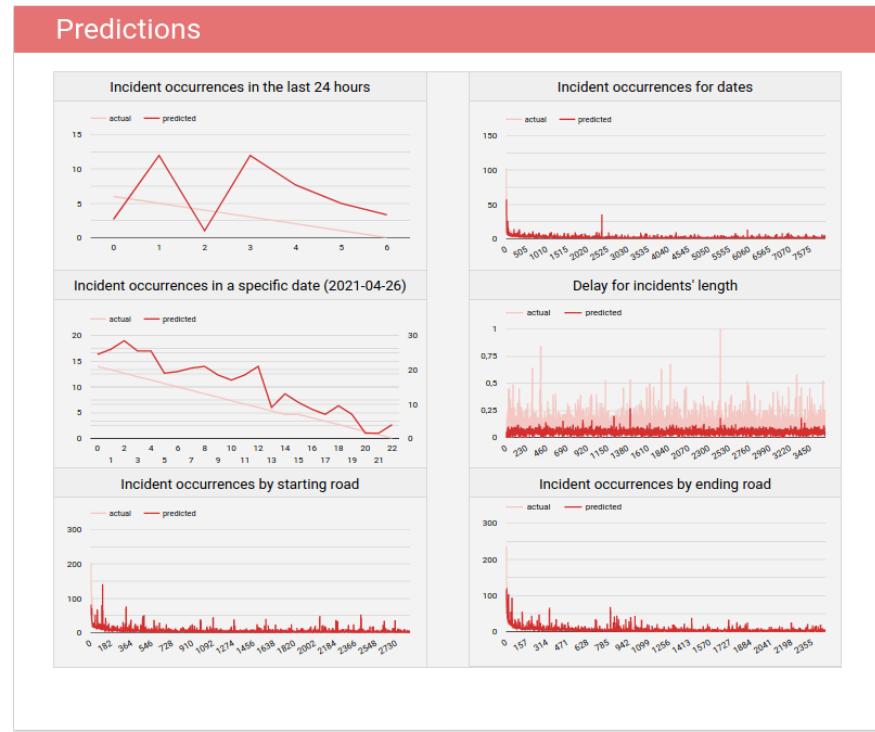


Figura 5.3: Página 3 - Previsões

Nesta terceira página são apresentados os gráficos implementados a partir do modelo descrito na secção 4. Para isso, no código desenvolvido, são exportados ficheiros CSV de cada um dos dados obtidos na aplicação do modelo KNN.

Através da visualização de todos estes dados, podemos tirar várias conclusões, nomeadamente que a maior parte dos incidentes tem início em Alfragide (Circular Regional Interna De Lisboa/A36) e fim em Buraca (A37) (Circular Regional Interna De Lisboa/A36), ocorre às 07:00 horas, provoca um elevado atraso, *etc.*

6 Aplicação Web

Nesta secção, iremos explicar a aplicação *web* desenvolvida, passando pela demonstração de cada uma das respetivas páginas. Esta aplicação pode ser consultada em <https://sites.google.com/view/incidenttracker>.

Desde já, é importante referir que, para desenvolver a aplicação, optámos por recorrer à ferramenta Google Sites, devido ao facto de esta não apresentar custos associados, facilitar bastante o processo de desenvolvimento, resultando num baixo tempo de execução e permitir integrar projetos realizados no Google Data Studio.

6.1 Home

Na Figura 6.1, é possível observar a página inicial da nossa aplicação que intitulámos de Incident Tracker.

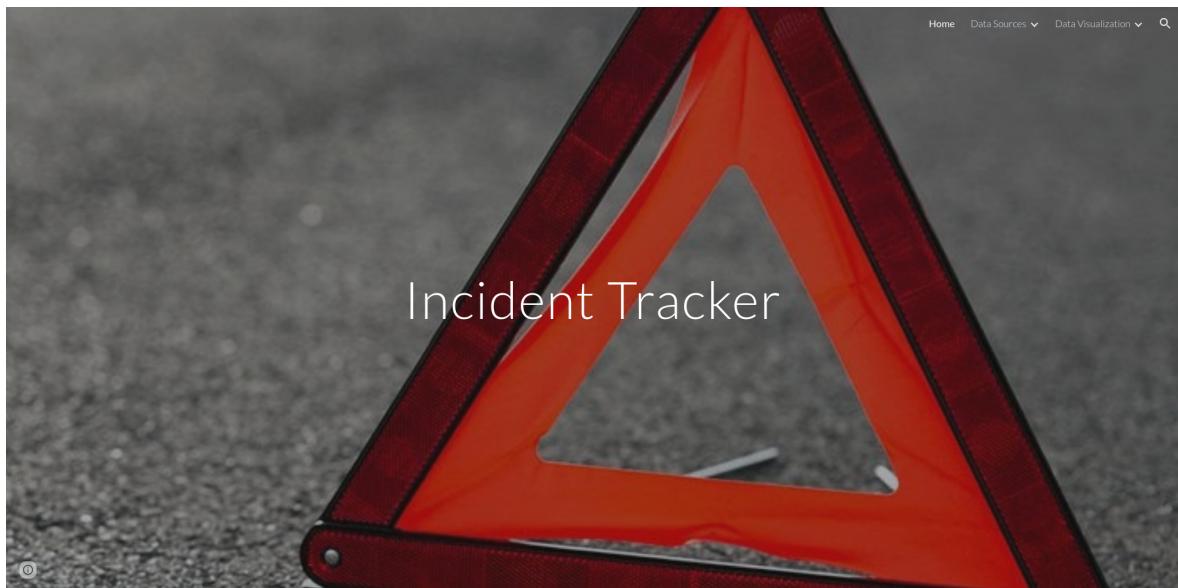


Figura 6.1: *Home*

6.2 Data Sources

Na Figura 6.2, é possível observar a página que nos irá redirecionar para cada uma das fontes de dados (neste caso, ficheiro `incidents.csv`, ficheiro `events.csv` e ficheiro `coordinates.csv`).

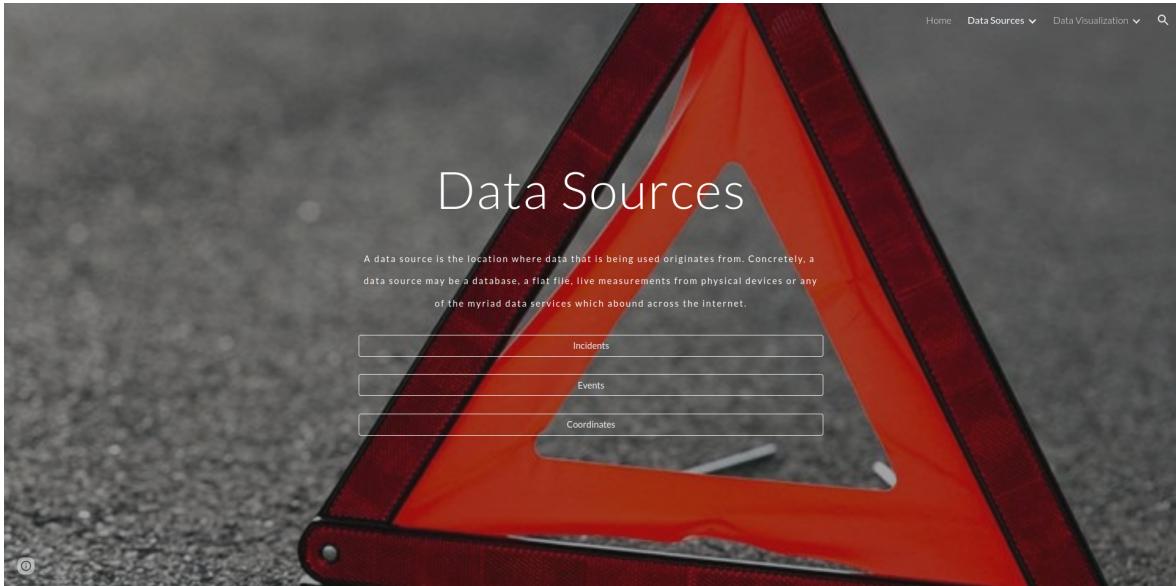


Figura 6.2: *Data Sources*

6.3 Incidents

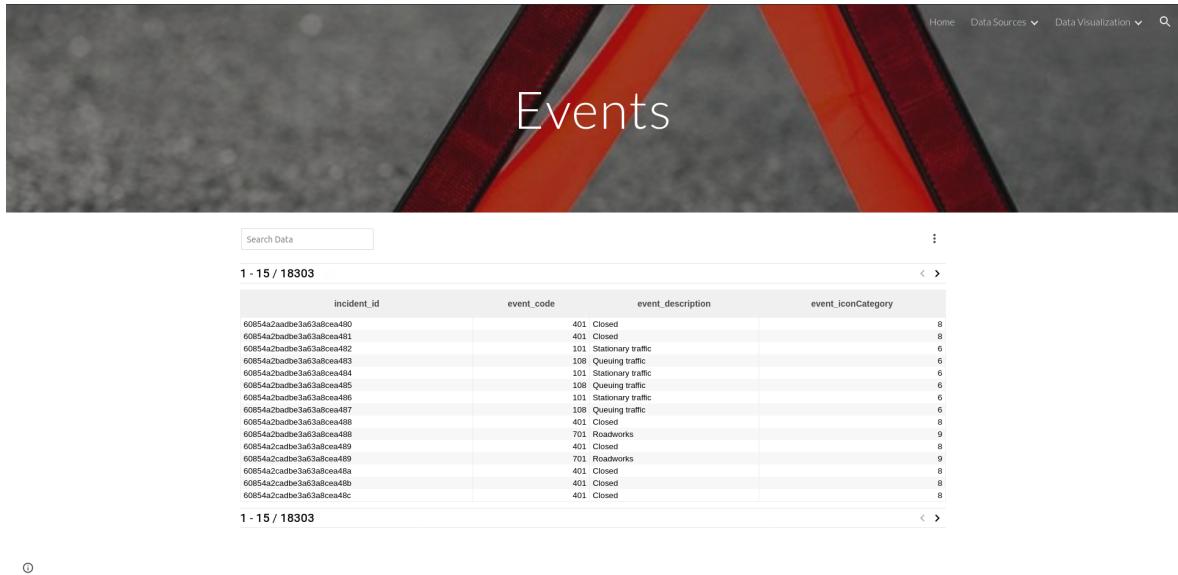
A página que podemos observar na Figura 6.3, apresenta uma tabela com os dados de uma das fontes de dados (neste caso, ficheiro `incidents.csv`). Para produzir a tabela em questão, simplesmente importámos o ficheiro mencionado para o Google Sheets e daí para o *site* Awesome Table. É importante mencionar que, para a tabela ter a funcionalidade de pesquisa, foi necessário realizar um pré-processamento que consistiu na concatenação das colunas. O resultado foi colocado numa nova coluna designada por **Search Data** que, por sua vez, contém o filtro **StringFilter-Hidden** não só para filtrar as colunas que correspondem à pesquisa, mas também para tornar essa coluna invisível.

1 - 15 / 18052						
<code>_id</code>	<code>type</code>	<code>properties_id</code>	<code>properties_iconCategory</code>	<code>properties_magnitudeOfDelay</code>	<code>properties_startTime</code>	<code>properties_end</code>
60854a2aadbe3a63a8cea480	Feature	b02cd3b91efabed2e3d945b35bb3ac9d9	8	4	2020-03-30T14:29:30Z	
60854a2aadbe3a63a8cea481	Feature	79b7ffce40e6ac525d05295547e134c3	8	4	2021-04-22T21:07:30Z	
60854a2aadbe3a63a8cea482	Feature	dc03a76de0a03ab58636d17r0275e991	6	3	2021-04-25T10:34:30Z	2021-04-25T11:1
60854a2aadbe3a63a8cea483	Feature	3fd6d8ff0922eb647fe3e782b6c6bef7	6	2	2021-04-25T10:48:00Z	2021-04-25T11:1
60854a2aadbe3a63a8cea484	Feature	6da03a2343c1ea16df542dcx0464705	6	3	2021-04-25T09:42:30Z	2021-04-25T11:1
60854a2aadbe3a63a8cea485	Feature	5ac8335a95b104be4bd765a00cbc3fb	6	2	2021-04-25T09:10:00Z	2021-04-25T11:1
60854a2aadbe3a63a8cea486	Feature	3472a310821420ec9e108a4202020	6	3	2021-04-25T08:51:30Z	2021-04-25T11:1
60854a2aadbe3a63a8cea487	Feature	177779c9945aa14bfaf5c514442e107f	6	2	2021-04-25T06:51:30Z	2021-04-25T11:1
60854a2aadbe3a63a8cea488	Feature	a5c3f205e271d2b3ab57c0872376756	8	4	2021-04-25T07:00:00Z	2021-04-25T19:0
60854a2cadbe3a63a8cea489	Feature	144d8f31eb2184db8e8ca2562c08b0	8	4	2021-04-25T07:00:00Z	2021-04-25T19:0
60854a2cadbe3a63a8cea48a	Feature	46c12d6c526e3df14f054779429d517	8	4	2020-06-01T11:20:30Z	
60854a2cadbe3a63a8cea48b	Feature	23effe35fbeda587487174ae03d4fcf9	8	4	2020-06-01T11:03:30Z	
60854a2cadbe3a63a8cea48c	Feature	a79b0dd6e55c2bc7b62c1c5e4cc2a2	8	4	2021-01-25T10:00:00Z	
60854a2cadbe3a63a8cea48d	Feature	2a65c7b5e447ad617658789687201o0e	8	4	2021-04-04T07:00:00Z	2021-05-31T19:0
60854a2cadbe3a63a8cea48e	Feature	8ca65e281aeb23f3c36b999e255252a3	8	4	2021-01-11T12:12:30Z	

Figura 6.3: *Incidents*

6.4 Events

Na Figura 6.4, é possível observar a página que apresenta uma tabela (desenvolvida da mesma forma que explicámos na secção 6.3) com os dados de uma das fontes de dados (neste caso, ficheiro `events.csv`).



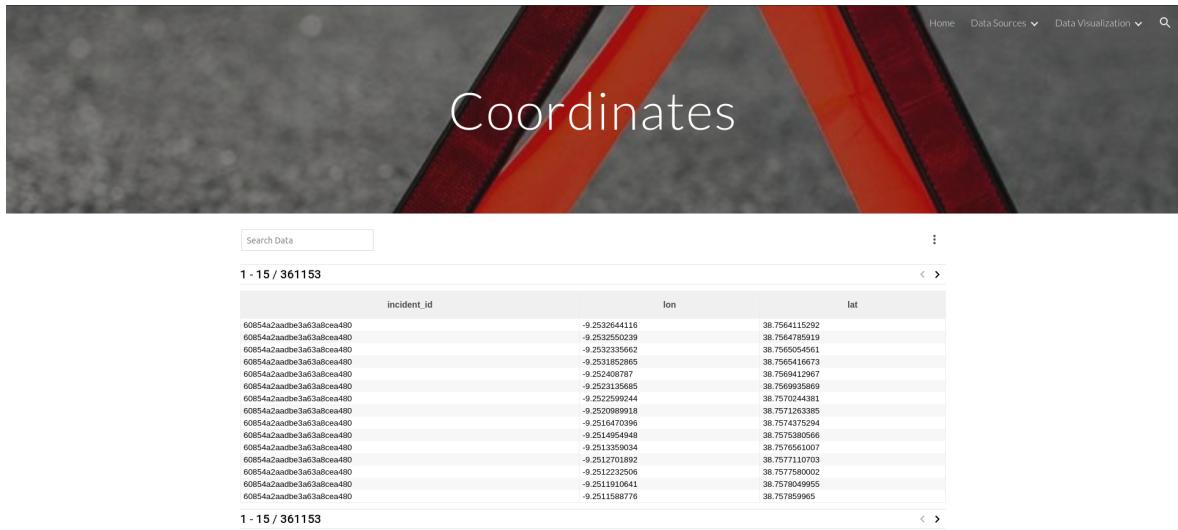
The screenshot shows a data visualization interface with a dark background featuring a red diagonal stripe pattern. At the top right are navigation links: Home, Data Sources, Data Visualization, and a search icon. Below the header is a large white text area containing the word "Events". Underneath this is a table with a light gray header row and white data rows. The table has four columns: incident_id, event_code, event_description, and event_iconCategory. The data consists of 15 rows of event records. At the bottom of the table is a footer row with the text "1 - 15 / 18303".

incident_id	event_code	event_description	event_iconCategory
608542aaadbe3af3a8cea480	401	Closed	8
608542aaadbe3af3a8cea480	401	Closed	8
608542aaadbe3af3a8cea482	101	Stationary traffic	6
608542aaadbe3af3a8cea483	108	Queuing traffic	6
608542aaadbe3af3a8cea484	101	Stationary traffic	6
608542aaadbe3af3a8cea485	108	Queuing traffic	6
608542aaadbe3af3a8cea486	101	Stationary traffic	6
608542aaadbe3af3a8cea487	108	Queuing traffic	6
608542aaadbe3af3a8cea488	401	Closed	8
608542aaadbe3af3a8cea489	701	Roadworks	8
608542aaadbe3af3a8cea489	401	Closed	8
608542aaadbe3af3a8cea489	701	Roadworks	9
608542aaadbe3af3a8cea488	401	Closed	8
608542aaadbe3af3a8cea488	401	Closed	8
608542aaadbe3af3a8cea48c	401	Closed	8

Figura 6.4: *Events*

6.5 Coordinates

Na Figura 6.5, é possível observar a página que apresenta uma tabela (desenvolvida da mesma forma que explicámos na secção 6.3) com os dados de uma das fontes de dados (neste caso, ficheiro `coordinates.csv`).



The screenshot shows a data visualization interface with a dark background featuring a red diagonal stripe pattern. At the top right are navigation links: Home, Data Sources, Data Visualization, and a search icon. Below the header is a large white text area containing the word "Coordinates". Underneath this is a table with a light gray header row and white data rows. The table has three columns: incident_id, lon, and lat. The data consists of 15 rows of coordinate records. At the bottom of the table is a footer row with the text "1 - 15 / 361153".

incident_id	lon	lat
608542aaadbe3af3a8cea480	-9.2532644116	38.7564115292
608542aaadbe3af3a8cea480	-9.2532550239	38.7564785919
608542aaadbe3af3a8cea480	-9.2532335662	38.7565054961
608542aaadbe3af3a8cea480	-9.2531852865	38.7565416673
608542aaadbe3af3a8cea480	-9.252409787	38.7569412967
608542aaadbe3af3a8cea480	-9.2523135685	38.7569935869
608542aaadbe3af3a8cea480	-9.2523045244	38.7571205311
608542aaadbe3af3a8cea480	-9.2523089918	38.7571205395
608542aaadbe3af3a8cea480	-9.2516472396	38.7571375294
608542aaadbe3af3a8cea480	-9.2514954948	38.7575380566
608542aaadbe3af3a8cea480	-9.2513359034	38.7576561007
608542aaadbe3af3a8cea480	-9.2513701892	38.7577110703
608542aaadbe3af3a8cea480	-9.2512332506	38.7577580002
608542aaadbe3af3a8cea480	-9.2511910641	38.7578049995
608542aaadbe3af3a8cea480	-9.2511588776	38.757859965

Figura 6.5: *Coordinates*

6.6 Data Visualization

Na Figura 6.6, é possível observar a página que nos irá redirecionar para os gráficos produzidos (neste caso, para os gráficos de geovisualização, para os gráficos de linhas/circulares/ *etc.* e para os gráficos com

as previsões).

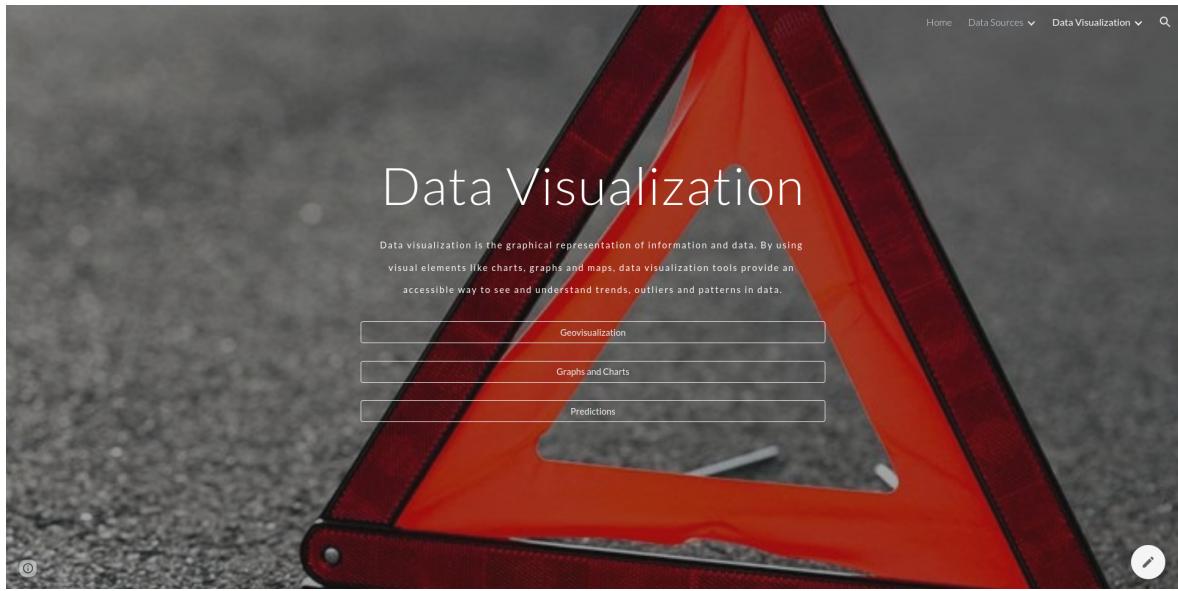


Figura 6.6: *Data Visualization*

6.7 Geovisualization

Na Figura 6.7, é possível observar a página com os gráficos de geovisualização. De notar que não foi possível embeber estes gráficos na aplicação *web*, uma vez que o Google Sites ainda não permite apresentar o Google Maps em projetos incorporados.

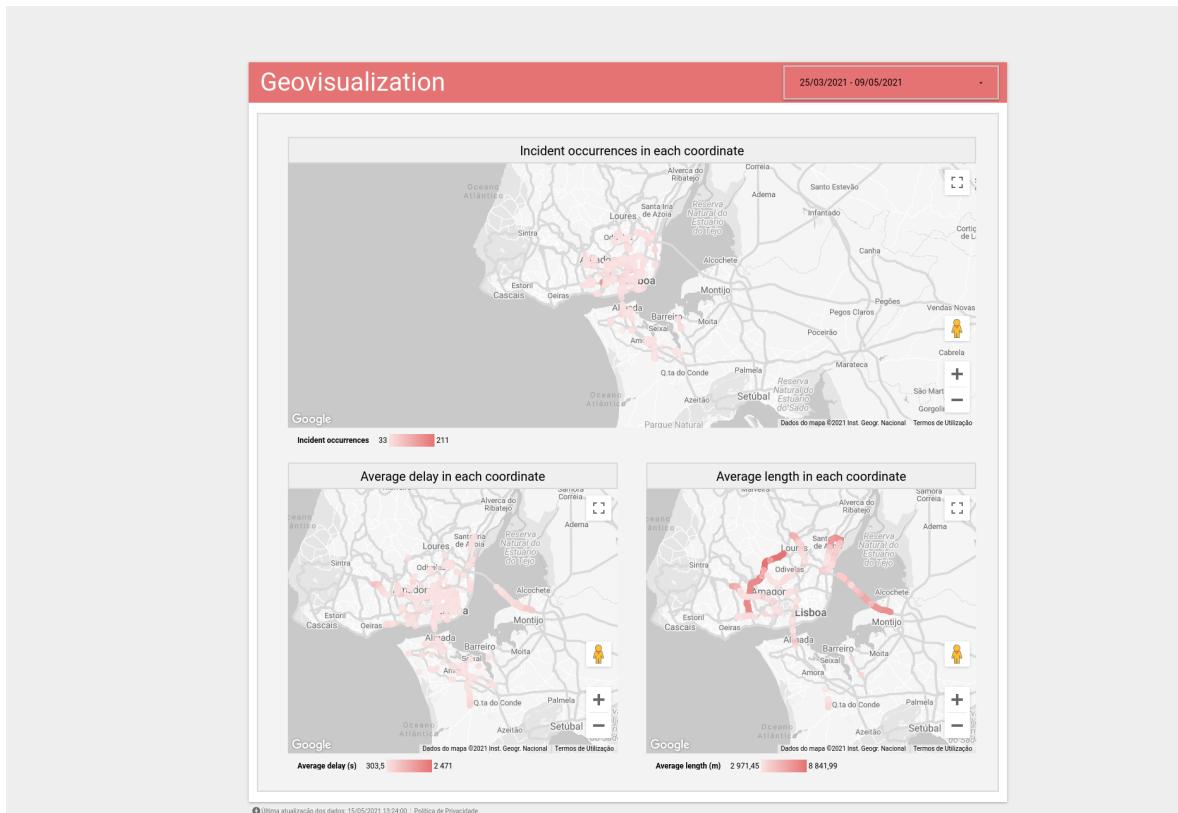


Figura 6.7: *Geovisualization*

6.8 Graphs and Charts

Na Figura 6.8, é possível observar a página com os restantes gráficos embbebidos na aplicação.



Figura 6.8: *Graphs and Charts*

6.9 Predictions

Na Figura 6.9, é possível observar a página com as previsões efetuadas, cujos gráficos também foram embbebidos na aplicação.



Figura 6.9: *Predictions*

6.10 Menu de pesquisa

Na Figura 6.10, é possível observar que a nossa aplicação possui um menu de pesquisa. Neste caso particular, pesquisámos a palavra *tools* e verificámos que esta está presente na página *Data Visualization*.

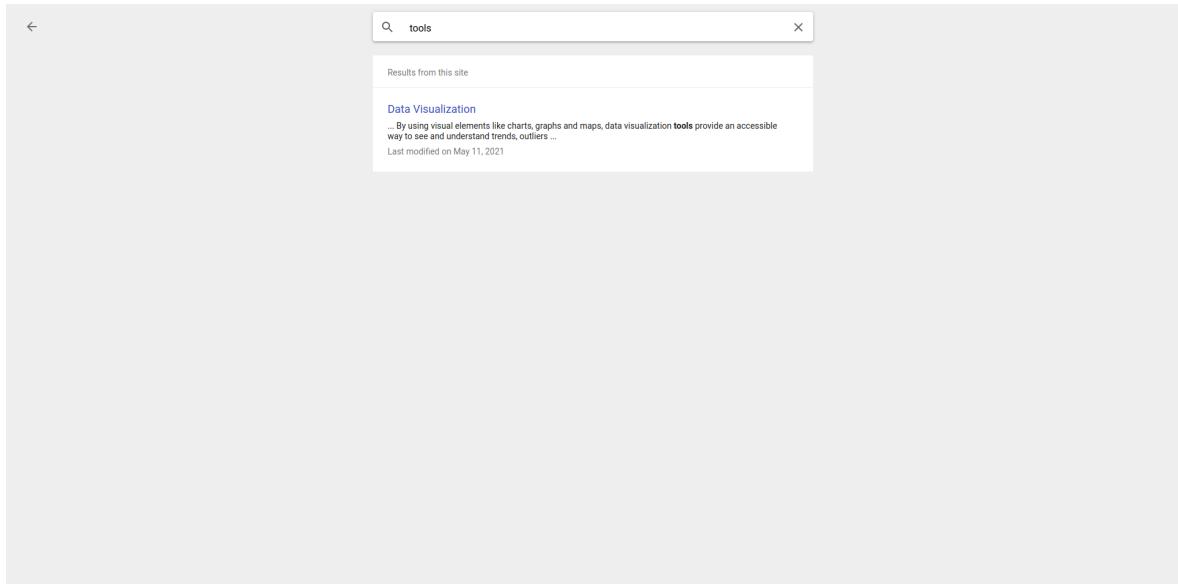


Figura 6.10: Menu de pesquisa

7 Conclusão

A realização deste projeto permitiu-nos ambientar a sensores virtuais utilizados nas mais variadas áreas, em particular na área da segurança rodoviária. Com isto, foi possível explorar novas APIs (neste caso, disponibilizadas pela TomTom) e compreender como é que a sensorização ambiente efetivamente funciona.

Durante a realização deste projeto recolhemos os dados presentes na API alusiva a incidentes rodoviários com recurso a um *script* que foi executado num servidor durante, sensivelmente, 2 semanas e armazenámos numa base de dados documental, tratámos os mesmos, transformando o paradigma documental em relacional, aplicámos modelos de *Machine Learning* para prever certos casos, visualizámos os dados com recurso a vários gráficos, como gráficos de linhas, gráficos circulares, *etc.* e apresentámos todo este processo numa aplicação *web*.

No processo de desenvolvimento do trabalho prático surgiram alguns dúvidas, tais como que ferramentas usar, qual a forma adequada de tratar os dados, que modelos de *Machine Learning* aplicar, que gráficos de visualização de dados apresentar, entre outras que, após alguma pesquisa e discussão, foram rapidamente ultrapassadas.

Em suma, todos os requisitos do trabalho foram cumpridos, mas acreditamos que ainda há espaço para várias melhorias, nomeadamente no que toca a recorrer a modelos de séries temporais, de modo a prever incidentes futuros, explorar mais cidades ou até mesmo analisar outras APIs disponibilizadas pela TomTom, para podermos tirar mais conclusões acerca de outros aspetos importantes na segurança rodoviária.