



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

LUCAS RODRIGUES CHAVES

**UM DATA WAREHOUSE DE DADOS ESPAÇO-TEMPORAIS PARA APOIO A
TOMADA DE DECISÃO NA ANÁLISE DE TRÁFEGO**

QUIXADÁ

2019

LUCAS RODRIGUES CHAVES

UM DATA WAREHOUSE DE DADOS ESPAÇO-TEMPORAIS PARA APOIO A TOMADA
DE DECISÃO NA ANÁLISE DE TRÁFEGO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informação
do Campus Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Sistemas de Informação.

Orientadora: Prof^a. M^a. Lívia Almada
Cruz

QUIXADÁ

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C439d Chaves, Lucas Rodrigues.
Um Data Warehouse de dados espaço-temporais para apoio a tomada de decisão na análise de tráfego /
Lucas Rodrigues Chaves. – 2019.
65 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,
Curso de Sistemas de Informação, Quixadá, 2019.
Orientação: Profa. Ma. Livia Almada Cruz.

1. Banco de Dados. 2. Armazém de dados. 3. Trajetória. 4. Informação. 5. Tomada de Decisão. I. Título.
CDD 005

LUCAS RODRIGUES CHAVES

UM DATA WAREHOUSE DE DADOS ESPAÇO-TEMPORAIS PARA APOIO A TOMADA
DE DECISÃO NA ANÁLISE DE TRÁFEGO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informação
do Campus Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Sistemas de Informação.

Aprovada em: ___/___/_____

BANCA EXAMINADORA

Prof^a. M^a. Lívia Almada Cruz (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Regis Pires Magalhães
Universidade Federal do Ceará (UFC)

Prof. Me. Jose Wellington Franco da Silva
Universidade Federal do Ceará (UFC)

À memória de Leandro Rodrigues Chaves. E à
minha família, por sua capacidade de acreditar e
investir em mim.

AGRADECIMENTOS

A Prof. Ma. Livia Almada Cruz por me orientar em meu trabalho de conclusão de curso. Sou grato pela paciência e atenção que me foi dada durante a realização deste trabalho e principalmente pela confiança depositada em mim, tenho certeza que sua passagem em minha vida contribuiu para que eu me tornasse uma pessoa melhor.

Ao Prof. Dr. Regis Pires Magalhães, Prof. Me. Tercio Jorge e Prof. Me. Jose Wellington Franco, pelas sugestões para melhorias no trabalho.

A toda minha família por acreditarem em mim e sempre me apoiarem, a minha força é a de vocês. Sou extremamente grato por tudo que fizeram por mim. Em especial a minha mãe Lucia Elena Chaves e minha avó Raimunda Barros Chaves por me acompanharem nas segundas-feiras de madrugada e por estarem comigo todos os momentos da minha vida, e à meu pai por depositar sua confiança em mim. A todos os outros avós, tios e tias, primos e primas por cada gesto de solidariedade para comigo.

Ao meu irmão, Leandro Rodrigues Chaves. Pela pessoa que foi e por tudo que me ensinou em vida, graças a sua existência eu pude me tornar a pessoa que sou hoje.

Aos meus amigos e colegas de curso: Antônio Alves, Bruno Pinho, Nathan Lima, Edir Lucas e João Paulo pelo companheirismo nessa jornada, vocês são parte desta história. Que o futuro de todos nós seja próspero.

A todos os meus amigos, colegas e pessoas que passaram em minha vida, que de certa forma contribuíram para que a conclusão deste trabalho se tornasse realidade.

Agradeço a todos os professores da UFC Campus Quixadá por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, pelos conhecimentos que me foram passados e por tanto que me fizeram evoluir.

Aos bibliotecários da Universidade Federal do Ceará Campus Quixadá, pela revisão e discussão da formatação utilizada neste trabalho.

“Os dias prósperos não vêm por acaso; nascem de muita fadiga e persistência.”

(Henry Ford)

RESUMO

A utilização de dados para a construção de informação que possa auxiliar a tomada de decisão se vê cada vez mais necessária nos dias atuais, ainda mais em ambientes que produzem um grande volume de dados. Este trabalho se trata de um estudo de caso exploratório, onde são mostradas as etapas realizadas para a construção e implementação de um modelo de dados multidimensional, para dados espaço-temporais formados a partir de registros efetuados por sensores de trânsito instalados em semáforos e radares de velocidade. O modelo permitirá a execução de consultas para a construção e descoberta de informações no sentido dar suporte e apoiar a tomada de decisão. O modelo desenvolvido permite a visualização de trajetórias construídas a partir dos dados coletados, além de outras informações de valor. E foi desenvolvido para usuários que possuam interesse em usufruir das informações sobre trânsito e tráfego de veículos, como órgãos públicos, autoridades de trânsito ou grupos de análise de dados.

Palavras-chave: Banco de Dados. Armazém de dados. Trajetória. Informação. Tomada de Decisão

ABSTRACT

The utilization of data for information construction that can help the making decision is increasingly necessary for current days, even more in environments that produce a lot of data. In this work is an exploratory case study, where will be shown the steps taken to accomplish the construction and implementation of a template of data multidimensional, for all spacetime formed from records made by sensors installed in traffic light and radars of high speed. Where the model will allow the execution of consults for the construction and discovery in the sence of giving support the support of making a decision. The model made enable the visualization of trajectories built form collected data, among other valuable information. And we made for the users that want to enjoy information about transit and vehicle traffic, as public organs, traffic authorities, or data analysis groups.

Keywords: Database. Data Warehouse. Trajectory. Information. Decision-making

LISTA DE FIGURAS

Figura 1 – Esquema Estrela	19
Figura 2 – Esquema Floco de Neve	19
Figura 3 – Processo ETL definido por Kimball e Caserta (2004)	20
Figura 4 – Primeiro Modelo Multidimensional Abstrato	28
Figura 5 – Primeiro Modelo Multidimensional Protótipo	29
Figura 6 – Tabelas povoadas com os dados da extração	30
Figura 7 – Tabelas com os dados tratados	31
Figura 8 – Atributos relacionados para junção das Tabelas	31
Figura 9 – Tabela união	32
Figura 10 – Modelo Multidimensional Abstrato Definitivo	34
Figura 11 – Modelo Multidimensional Definitivo	34
Figura 12 – Tabelas povoadas com os dados da extração	37
Figura 13 – Tabela registro_trajetoria	38
Figura 14 – Dados na tabela registro_trajetoria	39
Figura 15 – Modelo Multidimensional Final	39
Figura 16 – Registros por sensor	40
Figura 17 – Top 5 regiões em quantidade de registros	41
Figura 18 – Quantidade de registros por hora	41
Figura 19 – Quantidade de registros por dia	42
Figura 20 – Quantidade de registros por hora do dia	42
Figura 21 – Matriz origem-destino	43
Figura 22 – Mapa de trajetórias	44

LISTA DE TABELAS

Tabela 1 – Tabela Comparativa dos Trabalhos Relacionados	26
--	----

LISTA DE ABREVIATURAS E SIGLAS

AND	Associação Nacional de Detrans
CSV	<i>Comma-Separated Values</i>
IBPT	Instituto Brasileiro de Planejamento e Tributação
SGBD	Sistema de Gerenciamento de Banco de Dados

LISTA DE SÍMBOLOS

A_e	Área efetiva da antena
B	Largura de faixa em que o ruído é medido em Hertz
d	Distância em metros
E	Campo elétrico
FA	Fator da antena
Gr	Ganho de recepção
h	Altura efetiva ou comprimento efetivo de uma antena
I	Corrente elétrica
k	Constante de Boltzmann's
K	Eficiência de irradiação
M	Variação do patamar de ruído em função da RBW
N	Condutor de neutro
NF	Figura de ruído
N_i	Potência do ruído na entrada
N_o	Potência do ruído na saída
P	Potência
R	Resistência
S_i	Potência do sinal na entrada
S_o	Potência do sinal na saída
t	Tempo
V	Tensão
Z_L	Impedância da antena
Z_o	Impedância de referência (50Ω)
λ	Comprimento de onda
Γ	Coefficiente de reflexão

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo	16
1.2	Organização	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Data Warehouse	17
2.2	Modelos de Dados para Data Warehouse	18
2.2.1	<i>Modelo Multidimensional</i>	18
2.2.1.1	<i>Esquemas Multidimensionais</i>	18
2.3	Extração, Transformação e Carga	19
2.3.1	<i>Extração</i>	20
2.3.2	<i>Transformação</i>	20
2.3.3	<i>Carga</i>	21
2.4	Operações de um Data Warehouse	21
2.5	Trajetória	22
2.6	<i>Online Analytical Processing (OLAP)</i>	22
3	TRABALHOS RELACIONADOS	24
3.1	T-Warehouse: Visual OLAP analysis on trajectory data (LEONARDI <i>et al.</i>, 2010)	24
3.2	Mob-Warehouse: A semantic approach for mobility analysis with a Trajectory Data Warehouse	24
3.3	Approximate Aggregations in Trajectory Data Warehouses	25
3.4	Comparativo dos trabalhos relacionados	26
4	METODOLOGIA	27
4.1	Definição das perguntas norteadoras	27
4.2	Definição do Modelo Inicial: Tabelas Dimensões e Fato	28
4.2.1	<i>Modelagem do Modelo Multidimensional Protótipo</i>	28
4.3	Processo ETL	29
4.3.1	<i>Extração</i>	29
4.3.2	<i>Transformação</i>	30
4.3.3	<i>Carga</i>	32

4.4	Implementação de Consultas Sobre o Modelo	32
4.5	Avaliação de Resultados e Refinamento do Modelo	33
4.5.1	<i>Refinamento do modelo</i>	33
4.6	Definição de Tecnologias	35
4.7	Implementação	36
4.8	Extração	36
4.9	Transformação	37
4.10	Carga	39
5	RESULTADOS	40
6	CONCLUSÕES E TRABALHOS FUTUROS	45
	REFERÊNCIAS	46
	APÊNDICE A – CÓDIGOS SQL UTILIZADOS PARA PRODUÇÃO DO PRIMEIRO MODELO MULTIDIMENSIONAL	48
	APÊNDICE B – CÓDIGOS SQL UTILIZADOS PARA A ETAPA DE RE- FINAMENTO DO MODELO	53

1 INTRODUÇÃO

Um estudo do Instituto Brasileiro de Planejamento e Tributação (IBPT), feito sobre um levantamento de informações e publicado em 20/03/2018, mostrou que nesse respectivo ano havia no Brasil 65,8 milhões de veículos dos mais diversos tipos, entre leves, ônibus, caminhões e motocicletas, todos esses em efetiva utilização. Desse total, 41,2 milhões são automóveis (62,65%), 7,0 milhões são comerciais leves (10,67%), 2 milhões são caminhões (3,09%), 376,5 mil são ônibus (0,57%) e 15,1 milhões são motocicletas (23,01%) (EMPRESÔMETRO, 2018).

A partir de informações disponibilizadas pelo site da Associação Nacional de Detrans (AND), só no estado do Ceará no ano de 2019 com uma quantidade de 184 municípios foi contabilizado um total de 3.049.845 veículos automotores registrados. Com uma população total de 9.020.460 habitantes e 2.045.296 condutores habilitados temos uma estimativa aproximada de 1 veículo a cada 3 habitantes e de 1 veículo a cada 4,4 habitantes com habilitação (ASSOCIAÇÃO NACIONAL DOS DETRANS, 2019).

Do total de unidades do estado do Ceará, até março de 2016 36,2% dessas se concentraram apenas no município Fortaleza, indicando um enorme volume de veículos transitando pela cidade (DIARIO DO NORDESTE, 2016). Tais ruas e avenidas que são utilizadas para locomoção desses veículos possuem diversos semáforos e radares de velocidade com sensores de trânsito instalados, sensores esses que captam informações a todo instante e acabam gerando milhares de Gigabytes de dados que são guardados em bancos de dados e que permanecem por muito tempo sem serem utilizados.

Sobre esse enorme volume de dados que é produzido e armazenado, pode-se chegar a conclusão de que temos muitos dados, mas pouca informação. Dados que não são transformados em informação, não servem para muita coisa. A partir disso e diante do cenário atual, percebe-se a importância e necessidade de extrair valor sobre esses dados produzidos por sensores de trânsito, avaliando as informações de tráfego e mobilidade urbana. Portanto, levanta-se a questão: A partir dos dados gerados por sensores de velocidade e semafóricos podemos extrair alguma informação que gere valor ou auxilie a tomada de decisão?

Parte-se da hipótese que os dados são obtidos pela utilização de sensores que monitoram a movimentação dos veículos em vias urbanas. Cada sensor identifica a passagem de um veículo coletando a sua placa como um identificador do veículo, que antes de ser persistida no banco de dados é criptografada, o local, data e horário da passagem. Deseja-se responder a questões como: Quais as regiões tem tráfego mais intenso e quais horários e dias? Como se

dá o fluxo de veículos entre diferentes regiões e em quais dias e horários? Essas informações podem ser utilizadas por exemplo pela prefeitura da cidade para planejamento urbano e análise de tráfego. Assim, neste trabalho será desenvolvido um *Data Warehouse* para dar suporte a o processamento e análise dos dados provenientes desses sensores de trânsito, assim como para facilitar a tomada de decisão.

1.1 Objetivo

O objetivo geral deste trabalho é desenvolver um *Data Warehouse* para processar e analisar dados coletados a partir dos sensores de tráfego (de velocidade e semaforicos) instalados em pontos fixos da rede de ruas da cidade de Fortaleza-CE. Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Propor e desenvolver um modelo multidimensional espaço-temporal para dados de trânsito coletados de sensores que contemple diferentes granularidades de tempo e espaço.
- Construir um *Data Warehouse* que implemente o modelo proposto para persistir os dados coletados.
- Utilizar uma ferramenta que implemente consultas sobre o modelo multidimensional construído, extraindo informações de tráfego da cidade com elementos visuais intuitivos.

1.2 Organização

O Capítulo 2 apresenta a fundamentação teórica do trabalho. A metodologia é apresentada no Capítulo 4. No Capítulo 5 são apresentados os resultados deste trabalho e no Capítulo 6 a conclusão e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão abordados os principais conceitos relacionados a este trabalho.

2.1 Data Warehouse

Partindo da evolução das organizações, surgiu da necessidade de integrar dados corporativos espalhados em diferentes máquinas e sistemas operacionais, para torná-los acessíveis a todos os usuários dos níveis decisórios. Dessa forma, surgiu o conceito de *Data Warehouse* (DW) que segundo Inmon (1997), o definiu como: “Um conjunto de dados com informações corporativas integradas, não voláteis, orientados pelos assuntos principais da empresa e variantes no tempo, cujo objetivo é apoiar a tomada de decisão” (INMON, 1997).

Com base no que é descrito por Elmasri e Navathe (2010) os *Data Warehouses* oferecem armazenamento, funcionalidades e responsividade às consultas além das capacidades dos bancos de dados orientados a transação. Os *Data Warehouses* são otimizados para recuperação de dados e não para processamento de transações de rotina. Eles dão suporte a demandas de alto desempenho sobre os dados e informações de uma organização. Oferecendo acesso a dados para análise complexa e descoberta de conhecimento. Tendo assim uma característica distintiva de servir principalmente para aplicações de apoio à decisão.

Em comparação com os bancos de dados transacionais, os *Data Warehouses* são não voláteis. Isso significa que as informações no *Data Warehouse* mudam com muito menos frequência (apenas para correção) e podem ser consideradas informações que não são de tempo real com atualizações periódicas. Um *Data Warehouse* é frequentemente depósito de dados integrados de múltiplas fontes, processados para armazenamento em um modelo multidimensional.

As atualizações no *Data Warehouse* são tratadas pelo seu componente de aquisição do DW que é chamado de processo ETL do inglês *Extract, Transform and Load*, que oferece todo o pré-processamento exigido. O processo ETL está descrito na Seção 2.3.

Além disso, o desenvolvimento e a implantação deste ambiente envolvem a integração de dados de diversas fontes e sua transformação em informações consistentes e de qualidade, para permitir seu posterior emprego pelo usuário final no suporte à tomada de decisão (CAMPOS; BORGES, 2002).

Logo, como descrito por Machado (2000), o *Data Warehouse* é um armazém de dados históricos cuja finalidade é apresentar informações que permitam identificar indicadores e

a evolução destes ao longo do tempo.

2.2 Modelos de Dados para Data Warehouse

Nesta seção, é apresentado e discutido o modelo de dados multidimensional, o qual é o modelo utilizado para a construção de um *Data Warehouse*. São também apresentados diferentes esquemas do modelo multidimensional.

2.2.1 Modelo Multidimensional

Segundo Kimball e Caserta (2004), os modelos de dados dimensionais são as estruturas de dados mais populares para consulta e análise de usuários finais, pois são simples de criar, extremamente estáveis na presença de ambientes de dados variáveis e são a estrutura de dados mais rápida para a consulta de banco de dados.

Em Kimball e Ross (2013), é chegada a conclusão que a modelagem dimensional se provou ser a abordagem mais previsível e econômica para a construção de *Data Warehouses* e demonstração de dados analíticos, pois aborda conjuntamente dois requisitos primordiais para um projeto de *Data Warehouse*, que é o fornecimento de dados que sejam compreensíveis para os usuários corporativos e proporcionar desempenho de consulta rápida.

Modelos multidimensionais tiram proveito dos relacionamentos inerentes nos dados para preencher os dados em matrizes multidimensionais. Os modelos multidimensionais atendem prontamente a visões hierárquicas no que é conhecido como exibição *roll-up* ou exibição *drill-down*, que serão mais detalhados na seção 2.4. O modelo de armazenamento multidimensional envolve dois tipos de tabelas: tabelas de dimensão e tabelas de fatos (ELMASRI; NAVATHE, 2010).

Uma tabela de dimensão consiste em tuplas de atributos da dimensão. Uma tabela fato pode ser imaginada como tendo tuplas um para cada fato registrado. Esse fato contém alguma(s) variável(is) observada(s) e a(s) identifica com ponteiros para tabelas de dimensão. A tabela de fatos contém os dados e as dimensões identificam cada tupla nesses dados.

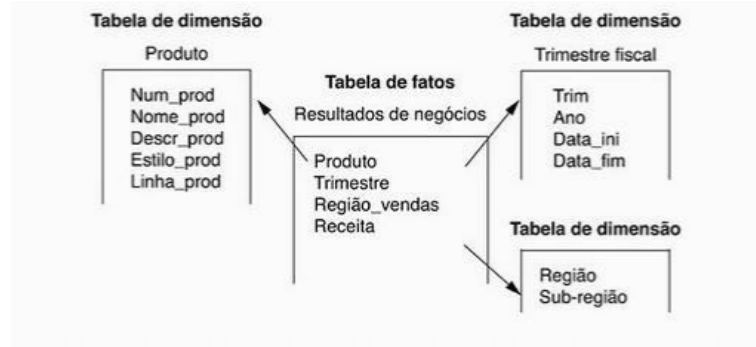
2.2.1.1 Esquemas Multidimensionais

Dois esquemas multidimensionais comuns são o Esquema Estrela e o Esquema Floco de Neve, definidos a seguir.

Definição 1 *Esquema de Estrela* é um esquema que consiste em uma tabela de fatos com uma única tabela para cada dimensão.

O esquema de estrela é representado na Figura 1.

Figura 1 – Esquema Estrela

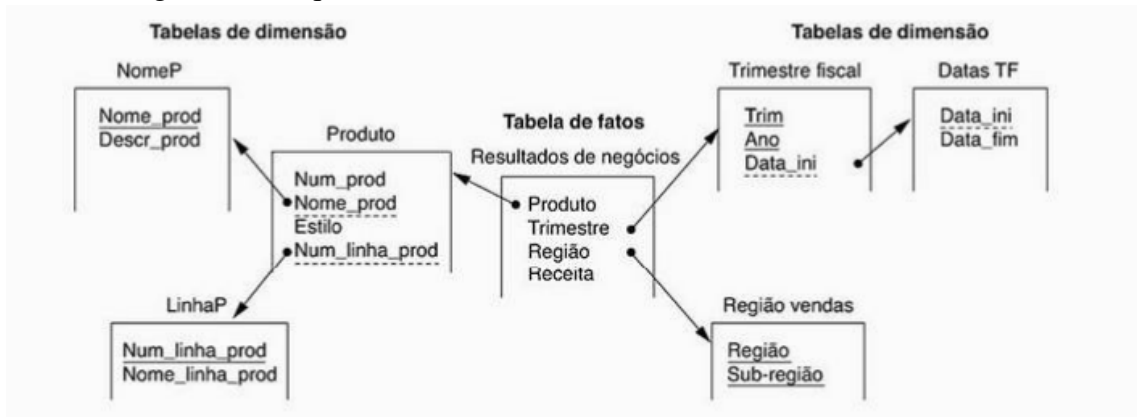


Fonte: (ELMASRI; NAVATHE, 2010)

Definição 2 *Snowflake* é uma variação do esquema estrela em que as tabelas dimensões de um esquema estrela são organizadas em uma hierarquia ao normalizá-las.

O esquema floco de neve é representado na Figura 2.

Figura 2 – Esquema Floco de Neve



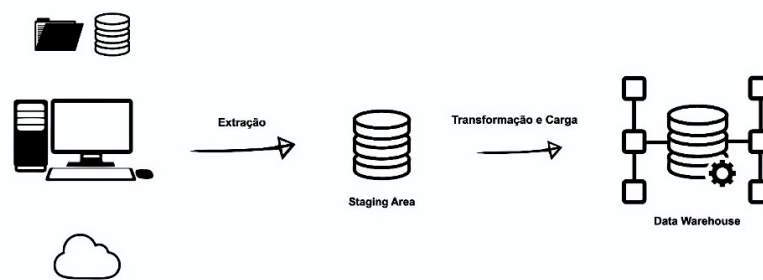
Fonte: (ELMASRI; NAVATHE, 2010)

2.3 Extração, Transformação e Carga

A abreviatura ETL (*Extract, Transform and Load*) vem do inglês e significa Extração, Transformação e Carga. Esse termo retrata todo o processo que envolve a coleta dos dados de fontes externas, transformação dos dados conforme as regras do negócio e o carregamento desses

dados dentro do *Data Warehouse*. Sendo uma das etapas mais importantes e o que consome mais tempo num projeto de *Data Warehouse* e segundo Kimball e Caserta (2004), esse pode consumir facilmente 70% dos recursos necessários para a implementação e manutenção de um projeto de *Data Warehouse*. Porque todo esse processo, a partir da coleta de dados, passando pela transformação até a inserção desses dados, é duradouro, complexo e crítico. A Figura 3 demonstra as etapas do ETL que serão descritas a seguir:

Figura 3 – Processo ETL definido por Kimball e Caserta (2004)



Fonte: Elaborado pelo autor

2.3.1 Extração

A extração dos dados dos sistemas de origem é o primeiro passo do processo ETL. Cada origem dos dados pode estar em um banco de dados diferente ou em plataformas diferentes (KIMBALL; CASERTA, 2004). Tais plataformas incluem planilhas eletrônicas, arquivos textos, entre outros. A extração dos dados deve ser feita levando em consideração os objetivos do *Data Warehouse*, pois a extração de dados desnecessários, além de ocupar um espaço considerável em disco, eleva o tempo de extração.

2.3.2 Transformação

Seguindo a ordem do processo ETL, essa é a segunda etapa a ser realizada. Que segundo Gonçalves (2003), os dados extraídos na etapa anterior são copiados para a *staging area*, que através de uma série de tratamentos será realizado o processo de transformação dos dados. Cujos objetivos são garantir a integridade dos dados através de programas ou rotinas especiais que tentam identificar anomalias e resolvê-las.

Resumindo, de acordo com Gonçalves (2003), o tratamento dos dados refere-se

então a limpeza ou filtragem dos dados, que busca garantir a integridade dos dados, identificando anomalias antes dos dados serem carregados no seu destino final. E a homogeneização dos dados, que serve para dar precisão aos dados, padronizar expressões, tipos de dados, entre outros. Deixando os dados em um estado consistente antes de estarem disponíveis no *Data Warehouse*. Esses dois processos são descritos a seguir:

- **Limpeza ou Filtragem dos dados:** Responsável pela correção de erros de digitação, violações de integridade, substituição de caracteres desconhecidos.
- **Homogeneização dos dados:** Responsável por colocar os dados em uma forma homogênea, determinando um único formato sem que ocorram conflitos de modelagem. Como por exemplo, padronizações de unidades de medida, de domínios e padronizações de tipos de dados.

2.3.3 Carga

Nessa etapa os dados já se encontram extraídos e limpos, prontos para serem carregados para um sistema final, por isso ela é a última fase do Processo ETL. Onde, Ponniah (2011) especifica e diferencia 3 tipos de cargas:

- a) **Carga Inicial:** Esse tipo de carga de dados, ocorre quando vamos inserir os dados no sistema de destino pela primeira vez.
- b) **Carga Incremental:** Quando que os dados do destino não estiverem atualizados, para que haja uma atualização de somente os dados em falta, usa-se esse tipo de carga de dados, porque esse utiliza uma forma de saber a última vez que os dados foram inseridos. Por isso somente os dados alterados depois dessa data serão carregados.
- c) **Carga Completa:** Os dados que se encontram no sistema de destino serão apagados (*Truncate* se for necessário) e carregados os que foram extraídos para esse destino. É como carregar todo o sistema de origem para dentro do sistema de destino, mas é importante não esquecermos que podemos ter várias fontes de dados.

2.4 Operações de um Data Warehouse

Nesta sessão serão listadas as seguintes possíveis operações padrão que podem ser executadas sobre um *Data Warehouse*.

- a) **Roll-up:** Os dados são resumidos com generalização cada vez maior (por exemplo,

semanal para trimestral para anual). A exibição *roll-up* sobe na hierarquia, agrupando em unidades maiores ao longo de uma dimensão (por exemplo somando dados semanais por trimestre por ano).

- b) **Drill-down:** Níveis cada vez maiores de detalhes são revelados (o complemento de *roll-up*: Uma exibição *drill-down* oferece a capacidade oposta, fornecendo uma visão mais detalhada. Como exemplo, dados sobre registros de vendas realizadas podem ser desagregadas de vendas do país por região e, depois, as vendas regionais por sub-região e também separando produtos por estilos.
- c) **Giro:** Uma rotação é realizada sobre as informações que estão sendo consultadas, trocando os atributos que correspondem as linhas e colunas.
- d) **Slice e Dice:** Operações de projeção que são realizadas nas dimensões, podendo entrar se aprofundar em um subconjunto de dados e expandir para um conjunto de dados maior.
- e) **Ordenação:** Os dados são ordenados por valor ordinal.
- f) **Seleção:** Os dados estão disponíveis por valor ou intervalo. Funciona como filtro sobre os atributos das dimensões.

2.5 Trajetória

Trajetória ou Trajetória Bruta, segundo Wagner *et al.* (2013), pode ser definido como: Uma trajetória T é uma lista ordenada de pontos espaço-temporais ou amostras $P_1, P_2, P_3, \dots, P_n$. Cada $P_i = (id, x_i, y_i, t_i)$ onde id é o identificador da trajetória, x_i, y_i são as coordenadas geográficas do ponto amostrado e t_i é o timestamp no qual o ponto foi coletado, com $T_1 \ll T_2 \ll T_3 \ll \dots \ll T_n$.

2.6 Online Analytical Processing (OLAP)

Segundo Anzanello (2007) Online Analytical Processing (*OLAP*) é o processamento analítico online dos dados que permite a visualização das informações a partir de muitas perspectivas diferentes, enquanto mantém uma estrutura de dados adequada e eficiente.

A visualização é realizada em dados agregados, e não em dados operacionais porque a aplicação OLAP tem por finalidade apoiar os usuários finais a tomar decisões estratégicas. Os dados são apresentados em termos de medidas e dimensão.

Na maioria das ferramentas de OLAP existentes no mercado atualmente é observado a existência de dois componentes: a ferramenta do administrador e a ferramenta do usuário

final. A ferramenta do administrador é usado para administrar e gerar os cubos de dados a serem acessados, enquanto o componente do usuário final, tem acesso aos dados para extraí-los de suas bases de dados, com os quais geram relatórios capazes de responder as suas questões gerenciais.

A ligação entre o *Data Warehouse* e o OLAP funciona de forma que, o *Data Warehouse* é utilizado para armazenar informações e o OLAP para recuperá-las, ambos são especializados para exercer suas funções de forma eficiente. As duas tecnologias são complementares de modo que um bom *Data Warehouse* é planejado com produção de relatórios em mente. Desta forma, para explorar o *Data Warehouse* completamente é necessário o OLAP que irá extrair e alavancar totalmente as informações nele contidas. (ANZANELLO, 2007).

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados alguns trabalhos relacionados destacando as semelhanças e diferenças com o desenvolvido neste trabalho.

3.1 T-Warehouse: Visual OLAP analysis on trajectory data (LEONARDI *et al.*, 2010)

O trabalho de Leonardi *et al.* (2010) apresenta um *framework* que transforma o modelo tradicional de cubo de dados em um DW de trajetória. Para realização do trabalho foram utilizados dados de dispositivos de reconhecimento de localização, como telefones celulares e dispositivos habilitados com GPS. O T-Warehouse é um sistema que incorpora todas as etapas necessárias para o DW de Trajetória, desde a reconstrução da trajetória e o processamento de ETL até a análise Visual OLAP sobre dados de mobilidade.

Na fase de pré-processamento é tratada a construção explícita das trajetórias, que são então armazenadas em um Banco de Dados de Objetos em Movimento (MOD), que oferece operações poderosas e eficientes para sua manipulação. Para a interatividade visual dos dados com o usuário, são desenvolvidas operações OLAP visuais, usando o Visual Analytics Toolkit (VAToolkit), um sistema interativo de informações geográficas baseado em Java. Este conjunto de ferramentas permite que um usuário possa ver dados geo-referenciados sobre um mapa e, também oferece funcionalidades para manipular dados temporais, usando gráficos ou animações, de acordo com o tipo de dados a analisar.

O foco do trabalho de Leonardi *et al.* (2010) é de ampliar as técnicas tradicionais de agregação para produzir informações sobre a trajetória e fornecer análises visuais de estilo OLAP. Portanto, o trabalho de Leonardi *et al.* (2010) se diferencia deste trabalho por seu foco ser na visualização, reconstrução e agregação de dados de trajetórias. Contudo, parte da proposta desse trabalho é de transformar dados de tráfego em informações valiosas que podem ser usadas para fins de tomada de decisão se assemelhando bastante com o trabalho atual, nesse ponto.

3.2 Mob-Warehouse: A semantic approach for mobility analysis with a Trajectory Data Warehouse

No trabalho de Wagner *et al.* (2013) é apresentado um modelo de Data Warehouse para trajetórias enriquecidas com semântica. Onde se é utilizado um grande conjunto de dados de trajetórias de carros, coletados de dispositivos com GPS. Nesse trabalho é considerado que uma

trajetória é uma lista ordenada pelo tempo, de pontos espaço-temporais ou amostras. Onde cada amostra possui um identificador da trajetória, coordenadas geográficas de latitude e longitude do ponto amostrado e o timestamp no qual o ponto foi coletado.

Enriquecimento semântico de dados de mobilidade é definido como dados de trajetórias que passam pelo processo de integração do conhecimento de domínio. Onde os pontos espaço-temporais que formam a trajetória são ligados aos dados do domínio de aplicação. A unidade de movimento é o ponto (espaço-temporal), dotado de várias dimensões semânticas incluindo a atividade, os meios de transporte e os padrões de mobilidade.

O trabalho de Wagner *et al.* (2013) busca a compreensão do comportamento dos objetos em movimento, com base nesse modelo implementado permitindo responder as perguntas que fornecem uma visão agregada de diferentes aspectos dos movimentos do usuário. Para alcançar o objetivo desse trabalho é utilizado o *framework* 5W1H, estabelecendo as perguntas (*Quem?*, *Onde?*, *Quando?*, *O que?*, *Por que?*, *Como?*). Que podem fornecer uma quantidade consistente de compreensão do contexto de uma circunstância. Cada questão narrativa do modelo 5W1H é mapeada para uma característica específica da trajetória. Podendo especificar informações contextuais sobre trajetórias e analisar os diferentes aspectos da “*história da mobilidade*” que o usuário “*está escrevendo*” com suas faixas.

Portanto, o foco do trabalho de Wagner *et al.* (2013) é a introdução de um modelo em que o componente espaço-temporal dos dados de trajetória é adequadamente integrado com informações relacionadas ao contexto, como meios de transporte, atividades executadas e padrões de mobilidade, se diferindo assim do trabalho atual. Este trabalho assemelha-se ao de Wagner *et al.* (2013) pois propõe a utilização de um DW para analisar dados de de tráfego e a vontade de promover informações de qualidade para dar ao usuário a flexibilidade de analisar o comportamento de acordo com vários pontos de vista.

3.3 Approximate Aggregations in Trajectory Data Warehouses

O trabalho de Braz *et al.* (2007) apresenta uma investigação sobre uma função agregada holística que retorna o número de trajetórias distintas ocorrendo em uma dada área espaço-temporal. Também é discutido como a tecnologia de data Warehouse pode ser usada para armazenar informações agregadas sobre trajetórias e executar operações OLAP sobre elas. O objetivo desse trabalho foi definir um modelo de data Warehouse simples para armazenar dados agregados sobre trajetórias, implementáveis usando sistemas de data Warehouse padrão

na indústria, além de implementar e avaliar a precisão de funções agregadas aproximadas e os erros induzidos no nível dos sub agregados na construção de um data Warehouse para dados espaço-temporais.

A solução proposta nesse trabalho utiliza o modelo definido por um esquema estrela onde as dimensões espaciais X, Y variam sobre os intervalos espaciais e a dimensão temporal T varia entre os intervalos temporais. Utilizou-se também da interpolação linear para inferir as localizações espaço-temporais de pontos intermediários, ocorrendo entre duas observações conhecidas na trajetória. Para a realização dos testes feitos neste trabalho foram empregados diversos conjuntos de dados sintéticos, obtidos por meio do gerador descrito por (BRINKHOFF, 2002), que produz trajetórias que representam os movimentos dos veículos em uma rede de estradas.

O trabalho de Braz *et al.* (2007) teve um enfoque nos problemas de pesquisa relacionados ao armazenamento e agregação de medidas holísticas. Os autores apresentaram os recursos e a complexidade das possíveis funções agregadas e delinearam possíveis maneiras de calcular de forma aproximada a presença agregada, além de propor maneira de calcular uma medida de presença distinta aproximada. Portanto, o trabalho de Braz *et al.* (2007) se relaciona com este trabalho pelo fato de utilizar funções agregadas e a construção de um data Warehouse para dados espaço-temporais. Entretanto, se destaca as diferenças com o trabalho atual no foco sobre as complexidades das funções agregadas, utilizar dados sintéticos e não definir níveis de granularidade diferentes na dimensão temporal.

3.4 Comparativo dos trabalhos relacionados

Essa seção apresenta a tabela de comparações de certos aspectos deste trabalho e os trabalhos relacionados apresentados. A Tabela 1 destaca a origem dos dados que os trabalhos usaram para serem realizados, responderá com sim ou não se o trabalho utiliza a tecnologia OLAP em seu desenvolvimento e também responde com sim ou não se os respectivos trabalhos utilizam diferentes níveis de granularidade temporal e espacial.

Tabela 1 – Tabela Comparativa dos Trabalhos Relacionados

	Origem dos Dados	OLAP	Granularidade Temporal	Granularidade Espacial
Leonardi <i>et al.</i> (2010)	GPS	Sim	Sim	Sim
Wagner <i>et al.</i> (2013)	GPS	Não	Sim	Sim
Braz <i>et al.</i> (2007)	Simulador de tráfego	Sim	Não	Sim
Trabalho atual	Sensores de trânsito	Sim	Sim	Sim

Fonte: Elaborado pela autor

4 METODOLOGIA

Este trabalho é produzido de forma exploratória e com avaliação dos resultados de forma qualitativa, uma vez que se busca a entrega de informações de qualidade relativas a seu domínio de aplicação.

Neste capítulo são apresentadas as etapas para o desenvolvimento deste trabalho. A seguir estão os passos necessários para a execução do trabalho:

1. Definição das perguntas norteadoras;
2. Definição do Modelo Inicial: Tabelas Dimensões e Fato;
3. Processo ETL;
4. Implementação de Consultas sobre o Modelo;
5. Avaliação de Resultados e Refinamento do Modelo;
6. Definição de Tecnologias;
7. Implementação do Sistema.

4.1 Definição das perguntas norteadoras

Nesta etapa com base na análise dos dados, foi realizado um levantamento de questões e informações nas quais o modelo proposto neste trabalho seja capaz de dar suporte à respostas.

Esta análise foi feita para definir as principais perguntas que o modelo proposto seja capaz de responder, e consultas que o mesmo seja hábil de dar suporte. As questões levantadas estão descritas a seguir.

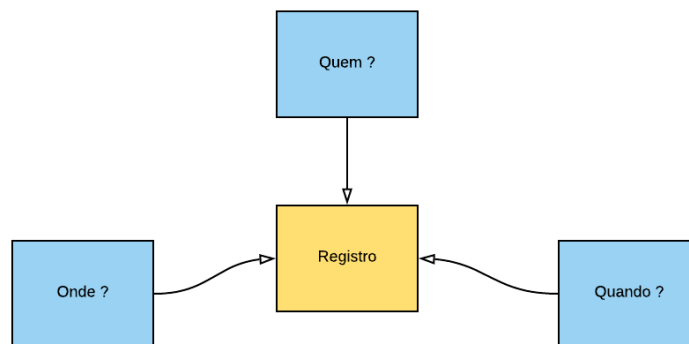
- Qual a porcentagem de registros por cada sensor instalado, baseado no total?
- Quais são os endereços com maior quantidade de registros?
- Como varia a quantidade de registros pelo tempo, mostrando quais são os horários com maior e menor quantidade de registros?
- Quantos registros no total foram efetuados em um determinado dia e hora?
- Como se dá o fluxo de veículos entre diferentes sensores e em quais dias e horários?
- Reconstruir as trajetórias feitas e descobrir quanto tempo elas duram.

4.2 Definição do Modelo Inicial: Tabelas Dimensões e Fato

Nesta etapa foi definido um primeiro modelo multidimensional com base nos dados que foram utilizados para o desenvolvimento do trabalho. Este modelo possui as tabelas dimensões e fato, conseguindo ser implementado em um banco de dados sendo utilizado a princípio com o intuito da efetuação da primeira carga de dados experimental, dando suporte às primeiras consultas.

Os dados utilizados foram coletados em forma de planilhas eletrônicas. Uma representando os dados mais gerais sobre os registros, com 15 atributos, e outra representando os sensores, com 7 atributos. Após o contato com os dados foi realizada uma análise sobre o domínio de aplicação desses dados. Depois dessa análise, foi feita a abstração do modelo, que está apresentado na Figura 4.

Figura 4 – Primeiro Modelo Multidimensional Abstrato



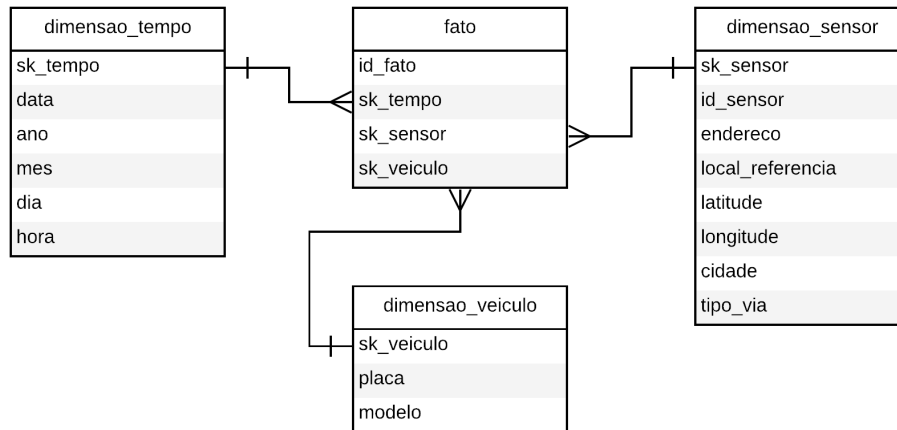
Fonte: Elaborado pelo autor

Esta modelagem foi feita baseada no Esquema Estrela. Como mostrado na Figura 1, é um conjunto de tabelas dimensão ligadas a uma única tabela fato. Cada tabela dimensão será encarregada de responder a uma pergunta entre: "Quem?", "Onde?" e "Quando?". As respostas dessas questões definem um registro de tráfego, que será representado pela tabela fato.

4.2.1 Modelagem do Modelo Multidimensional Protótipo

A partir de uma análise mais aprofundada sobre os dados e o modelo abstrato, foi feita a definição de um modelo real para ser desenvolvido e suportar a carga e consultas sobre tais dados. O modelo é apresentado na Figura 5 abaixo.

Figura 5 – Primeiro Modelo Multidimensional Protótipo



Fonte: Elaborado pelo autor

Neste modelo são mostrados os atributos que formam cada dimensão correspondente às respostas das perguntas do modelo multidimensional abstrato mostrado na Figura 4.

Quem? O veículo que está no registro, com placa anonimizada e modelo do veículo.

Onde? Local onde o foi efetuado o registro (local onde o sensor que efetuou o registro está localizado), essa dimensão tem como atributos um identificador único, endereço onde o sensor está localizado, um local de referência, latitude, longitude, cidade e o tipo da via onde está localizado.

Quando? Horário em que o registro foi efetuado, essa dimensão é carregada com os dados no formato de timestamp de quando foi efetuado o registro com ano, mês, dia, horas, minutos, segundos no formato “aa/mm/dd hh:mm:ss”. Que foi fatorado em atributos separados de ano, mês e dia e hora, aumentando o nível de granularidade.

4.3 Processo ETL

Nesta etapa foi executado o processo de Extração, Transformação e Carga sobre os dados no primeiro modelo definido neste trabalho.

4.3.1 Extração

A fase de extração começou a partir da coleta dos dados que estão em forma de planilhas eletrônicas no formato *Comma-Separated Values* (CSV). Foram criadas duas tabelas para serem populadas com os respectivos dados contidos nestes arquivos. Essa população

foi efetuada a partir da função *import* do DBeaver, um cliente SQL e uma ferramenta de administração de banco de dados. Essas tabelas estão representadas nas Figuras 7(a) e 7(b).

Figura 6 – Tabelas povoadas com os dados da extração

base_extracao	
base_data	
ABC base_cidade	
ABC base_endereco	
ABC base_branco	
ABC base_placa	
ABC base_modelo	
ABC base_cor	
ABC base_estado	
ABC base_dezena	
ABC base_unidade	
ABC base_situacao	
ABC base_ais	
ABC base_cidadex	
ABC base_tipovia	
ABC base_local_referencia	

sensores_extracao	
123 sensor_id	
123 sensor_id_2	
ABC sensor_endereco	
ABC sensor_cidade	
ABC sensor_tipovia	
ABC sensor_local_referencia	
123 sensor_latitude	
123 sensor_longitude	

(a) Tabela base_extracao

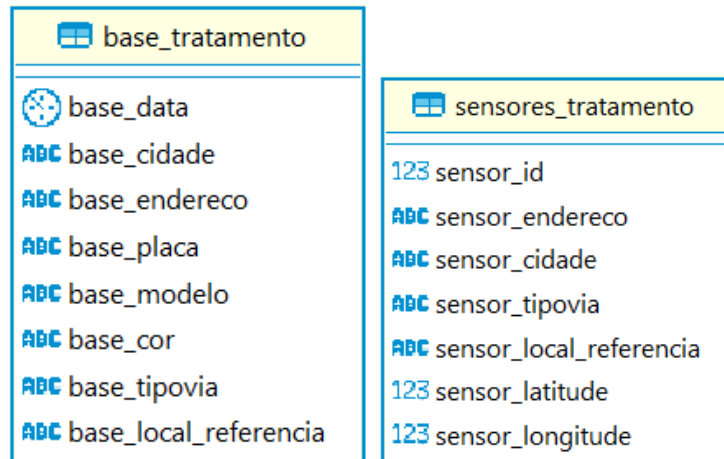
(b) Tabela sensores_extracao

Fonte: Elaborada pelo autor

4.3.2 Transformação

Na fase de transformação foram criadas duas tabelas para serem povoadas com os dados tratados a partir das tabelas mostradas na fase de extração. A inserção dos dados foi feita com a utilização do comando SQL 3 para popular a tabela base_tratamento com apenas os dados que serão usados nas próximas etapas e foram coletados no dia 01/09/2017 na cidade de Fortaleza e o comando 4 para popular a tabela sensores_tratamento com apenas os dados dos sensores instalados na cidade de Fortaleza. Condicionando os dados para que sejam utilizados apenas os de real interesse. Assim os próximos passos puderam ser realizados com dados mais adequados. Tais tabelas estão mostradas nas Figuras 8(a) e 8(b).

Figura 7 – Tabelas com os dados tratados



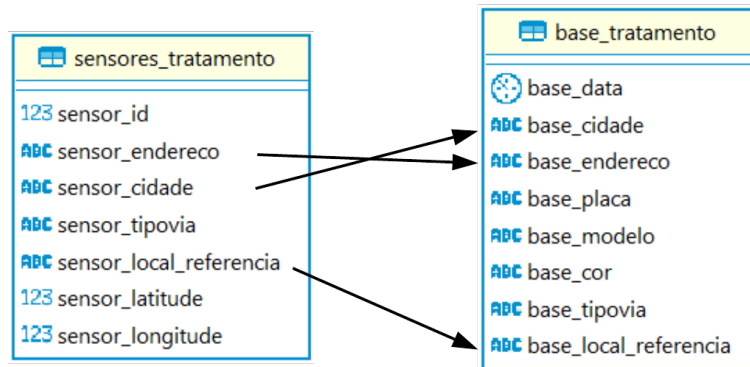
(a) Tabela base_tratamento

(b) Tabela sensores_tratamento

Fonte: Elaborada pelo autor

Após a criação destas tabelas, foi efetuada uma junção das duas tabelas produzidas para formar a tabela_master. Essa união foi feita a partir do relacionamento de 3 colunas que se repetiam nas duas tabelas, referentes a: Cidade, endereço e local de referencia de onde foi efetuado o registro.

Figura 8 – Atributos relacionados para junção das Tabelas



Fonte: Elaborado pelo autor

A tabela criada a partir desta junção representa todos os registros de forma completa, cada tupla caracteriza um registro que contem os dados sobre a passagem de um veiculo, por um local em uma determinada data e está representada na Figura 9.

Figura 9 – Tabela união

tabela_master	
123	id_master
base_data	
ABC	base_cidade
ABC	base_endereco
ABC	base_placa
ABC	base_modelo
ABC	base_cor
ABC	base_tipovia
ABC	base_local_referencia
123	sensor_id
ABC	sensor_endereco
ABC	sensor_cidade
ABC	sensor_tipovia
ABC	sensor_local_referencia
123	sensor_latitude
123	sensor_longitude

Fonte: Elaborado pelo autor

4.3.3 Carga

Nesta etapa foi realizada uma carga inicial nas tabelas dimensões e fato do modelo multidimensional mostrado na Figura 5. Utilizando dos dados já devidamente tratados da tabela geral. Cada dimensão foi carregada com os atributos necessários para a respectiva dimensão. Sendo adicionado um atributo identificado como *Surrogate Key* para cada tabela dimensão, onde esse atributo será incrementado a cada tupla carregada em sua respectiva tabela dimensão.

Por fim, foi realizada a carga da tabela fato, que através de comandos SQL apresentados no Apêndice A pôde ser povoada com o conjunto de *Surrogate Key's* das tabelas dimensão. Cada tupla da tabela fato é um conjunto de chaves que servem como ponteiros para os dados que representarão um registro nas tabelas dimensão. Todos os códigos utilizados nesta etapa estão no Apêndice A.

4.4 Implementação de Consultas Sobre o Modelo

Nesta etapa foram desenvolvidas consultas utilizadas sobre o primeiro modelo criado. O objetivo principal desta etapa foi execução destas consultas visando a identificação de quais informações eram possíveis extrair a partir do modelo multidimensional implementado,

informações essas que serviram de parâmetro de análise para a próxima etapa.

Após a execução das consultas desenvolvidas sobre modelo multidimensional criado.

Foram obtidos resultados capazes de responder a questões como:

- Quantos veículos passaram por cada sensor no total?
- Quantidade de registros total por mês, dia e hora.
- Quais foram os sensores pelos quais um veículo passou em um determinado dia ordenados pelo tempo?
- Quantos veículos passaram por cada sensor no total em um determinado ano, mês, dia, hora?

Essas foram apenas algumas questões destacadas que as respostas foram possíveis de serem extraídas a partir de consultas rápidas ao modelo multidimensional. Sendo que todas essas respostas podem ser ordenadas por níveis diferentes de granularidade do tempo. Todos os resultados desta etapa são apresentados neste repositório no GitHub ¹.

4.5 Avaliação de Resultados e Refinamento do Modelo

Nesta etapa foram efetuadas avaliações sobre o que foi produzido decorrente das etapas anteriores. Identificando questões de interesse e verificando a existência de erros, possíveis melhorias e mudanças no modelo. Foi feita uma reavaliação sobre o modelo, a organização dos dados e as consultas respondidas. A fim de encontrar informações de valor que ainda poderiam ser obtidas a partir desses dados.

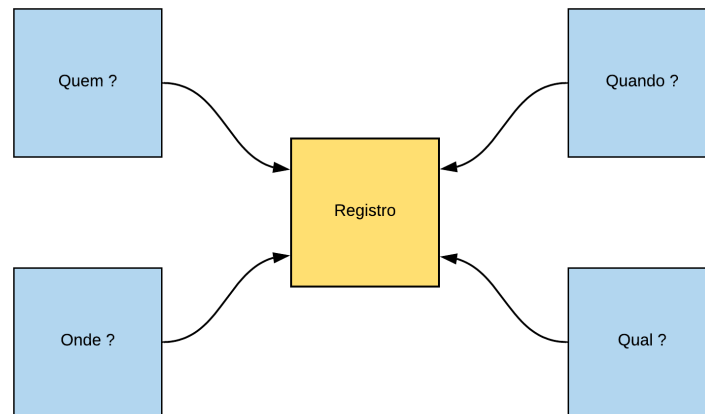
A partir desta etapa de avaliação, foram identificadas possíveis melhorias no processo *ETL*, mudanças nas tabelas dimensões e fato e a necessidade de uma nova tabela dimensão para que o modelo pudesse satisfazer algumas questões relacionadas a reconstrução de trajetórias.

4.5.1 Refinamento do modelo

Nesta etapa foram executadas as ações para a construção do modelo multidimensional definitivo deste trabalho. Começando pela abstração de um novo modelo que seja capaz de suportar todas as questões destacadas na Sessão 4.1. Este novo modelo continua com as questões mostradas na Figura 4, com a adição de uma nova dimensão correspondente a questão: "Qual?". Assim, as respostas dessas questões além de definirem um registro de tráfego, também pode determinar as trajetórias percorridas pelos veículos, este modelo está ilustrado na Figura 10.

¹ <<https://github.com/lucasrc98/TCC-Scripts>>

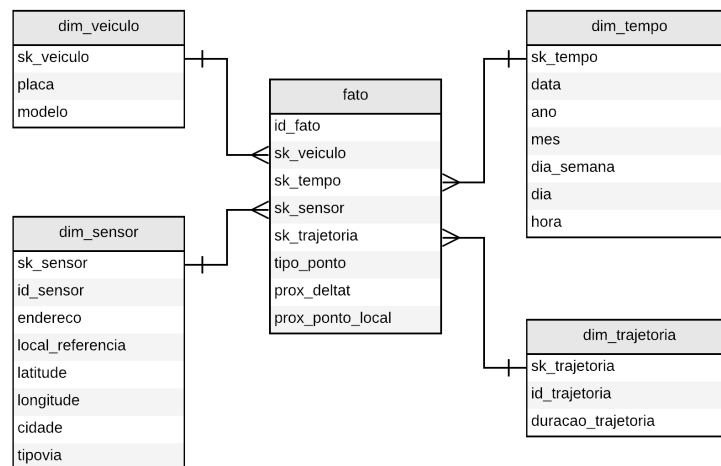
Figura 10 – Modelo Multidimensional Abstrato Definitivo



Fonte: Elaborado pelo autor

Com a elaboração deste novo modelo, se fez necessário adição de uma nova tabela dimensão, onde são armazenados os dados necessários para a reconstrução das trajetórias. A partir deste modelo abstrato mostrado na etapa anterior, foi feita a definição de um novo modelo real que foi desenvolvido para suportar a carga e consultas sobre tais dados. A representação deste modelo é apresentada na Figura 11.

Figura 11 – Modelo Multidimensional Definitivo



Fonte: Elaborado pelo autor

Neste modelo são mostrados os atributos que formam cada dimensão correspondente as respostas das perguntas do modelo multidimensional abstrato mostrado na Figura 10.

Quem? Respondida pela tabela dim_veiculo, que representa o veículo que está no registro, com placa anonimizada e modelo do veículo.

Onde? Respondida pela tabela dim_sensor, que representa o local onde o foi

efetuado o registro (local onde o sensor que efetuou o registro está localizado), essa dimensão tem como atributos um identificador único do sensor, endereço onde o sensor está localizado, um local de referência, latitude, longitude, cidade e o tipo da via onde está localizado.

Quando? Respondida pela tabela *dim_tempo*, que representa o horário em que o registro foi efetuado, essa dimensão é carregada com os dados no formato de timestamp de quando foi efetuado o registro com ano, mês, dia, horas, minutos, segundos no formato “aa/mm/dd hh:mm:ss”. Que foi fatorado em atributos separados de ano, mês, dia da semana, dia e hora.

Qual? Respondida pela tabela *dim_trajetoria*, que representando a trajetória percorrida por um veículo em um determinado intervalo de tempo, essa dimensão tem como atributos um identificador e a duração de tempo total de uma trajetória.

Todas as tabelas dimensões tem um atributo *Surrogate Key* com a abreviatura *sk*, este é um atributo identificador de cada linha de uma tabela dimensão, que é incrementado a cada registro adicionado a uma tabela dimensão.

Registro É representado pela tabela *fato*, com um identificador único para cada registro salvo, as *surrogate keys* de todas as tabelas dimensão, um atributo para classificar o tipo do ponto para cada registro efetuado com: inicial, intermediário ou final, um atributo que armazena o tempo que um veículo levou para chegar do ponto atual de um registro para o próximo registro e o local onde foi efetuado o próximo registro.

4.6 Definição de Tecnologias

Existem várias ferramentas que podem ser utilizadas para a construção de um modelo de dados e um ambiente de visualização de informações, na atualidade. O objetivo dessa etapa é escolher quais ferramentas serão utilizadas para a produção deste trabalho. Foi escolhido o Sistema de Gerenciamento de Banco de Dados (SGBD) PostgreSQL para o efetuar o armazenamento dos dados, também foi escolhido o DBeaver Community que é um cliente SQL e uma ferramenta de administração de banco de dados. Ambos são tecnologias com licença grátis.

Já para a construção e visualização das informações foi utilizado o Tableau Desktop. Esta ferramenta possui funcionalidades que se alinham com os objetivos do trabalho atual e possam auxiliar a satisfazer as necessidades levantadas na Sessão 4.1 e que possam proporcionar o desenvolvimento eficiente do resultado final deste projeto. Diferentemente das anteriores o Tableau Desktop não é de licença grátis para todos, mas é disponibilizado licenças gratuitamente

para quem comprova que é estudante.

4.7 Implementação

Nessa etapa ocorreu o desenvolvimento de um *Data Warehouse* para dados provenientes de sensores de tráfego de veículos e a utilização de uma ferramenta para a visualização e apresentação das informações extraídas do modelo multidimensional definido anteriormente neste trabalho. Para esta implementação foi aplicado o processo ETL utilizados os dados coletados do dia 06/09/2017 a 12/09/2017 na cidade de Fortaleza-CE. Todos os códigos produzidos nesta etapa do trabalho estão disponíveis neste repositório no GitHub ².

4.8 Extração

A princípio foram criados os *schemas Staging Area* e *Data Warehouse* para a separação das tabelas. A fase de extração começou a partir da coleta dos dados que estão em forma de planilhas eletrônicas no formato CSV. Foram criadas duas tabelas na *Staging Area* para serem povoadas com os respectivos dados contidos nestes arquivos. Essa população foi efetuada a partir da função *import* do DBeaver. Os códigos utilizados para a criação dessas tabelas são os itens 11, 12 no Apêndice B e elas estão representadas nas Figuras 13(a) e 13(b).

² <<https://github.com/lucasrc98/TCC-Scripts>>

Figura 12 – Tabelas povoadas com os dados da extração

registro_extracao	
base_data	
ABC base_cidade	
ABC base_endereco	
ABC base_branco	
ABC base_placa	
ABC base_modelo	
ABC base_cor	
ABC base_estado	
ABC base_dezena	
ABC base_unidade	
ABC base_situacao	
ABC base_ais	
ABC base_cidadex	
ABC base_tipovia	
ABC base_local_referencia	

sensor_extracao	
123 sensor_id	
123 sensor_id_2	
ABC sensor_endereco	
ABC sensor_cidade	
ABC sensor_tipovia	
ABC sensor_local_referencia	
123 sensor_latitude	
123 sensor_longitude	

(a) Tabela registro_extracao

(b) Tabela sensor_extracao

Fonte: Elaborada pelo autor

4.9 Transformação

Após os dados serem armazenados nas tabelas criadas na fase de extração, foi criada a tabela registro_trajetoria, que é necessária para armazenar os dados da união entre as tabelas registro_extracao e sensor_extracao com mais alguns atributos calculados que são utilizados para a reconstrução das trajetórias. Tais atributos são id_trajetoria que sera responsável por manter um mesmo número identificador para todos os registros de uma trajetória, pre_data que armazena a data e horário em que o registro de um mesmo veículo foi efetuado anteriormente do registro atual e prox_data que armazena a data e o horário do registro de um mesmo veículo efetuado após o atual. O atributo pre_deltat armazena o tempo de diferença entre o registro anterior de um mesmo veiculo para o atual e o prox_deltat é o tempo de diferença entre o registro para o próximo registro de um mesmo veículo. A tabela está representada pela Figura 13.

Figura 13 – Tabela registro_trajetoria

registro_trajetoria	
123	id_registro
123	id_trajetoria
🕒	base_data
ABC	tipo_ponto
ABC	base_cidade
ABC	base_endereco
ABC	base_placa
ABC	base_modelo
ABC	base_tipovia
ABC	base_local_referencia
123	sensor_id
ABC	sensor_endereco
ABC	sensor_cidade
ABC	sensor_tipovia
ABC	sensor_local_referencia
123	sensor_latitude
123	sensor_longitude
🕒	pre_data
🕒	pre_deltat
🕒	prox_data
🕒	prox_deltat

Fonte: Elaborado pelo autor

Antes de ser efetuada a inserção dos dados na tabela registro_trajetoria, foi criada uma função no banco de dados utilizando a linguagem plpgsql, como mostrado no item 16 no Apêndice B, esta função impede a inserção de registros duplicados, atribui o valor nulo para o atributo pre_data se não houver algum registro efetuado até 20 minutos antes do registro atual, atribui o valor nulo para o atributo prox_data se não houver algum registro efetuado até 20 minutos após o registro atual e impede a inserção de registros únicos. Também foi criada uma *trigger* que ativa essa função antes da inserção de cada linha na tabela registro_trajetoria mostrado no item 17.

Após isso foi criada mais uma função no banco de dados, que por sua vez é responsável por atribuir os valores para os identificadores das trajetórias e os tipos dos registros como: Inicial, intermediário e final. Ela faz isso baseando-se nos valores dos atributos pre_deltat e prox_deltat, para a utilização desta função foi criada a tabela temp_tb que armazena um número incremental que é utilizado para definir os identificadores das trajetórias. Também foi criada uma *trigger* que ativa esta função logo após cada inserção efetuada na tabela registro_trajetoria. Os códigos para a criação da função, da tabela temp_tb e da *trigger* mostradas estão nos itens 18,

20, 19 do Apêndice B respectivamente e a inserção foi feita a partir do código 21 e os dados na tabela registro_trajetoria ficaram organizados como mostrado na Figura 14.

Figura 14 – Dados na tabela registro_trajetoria

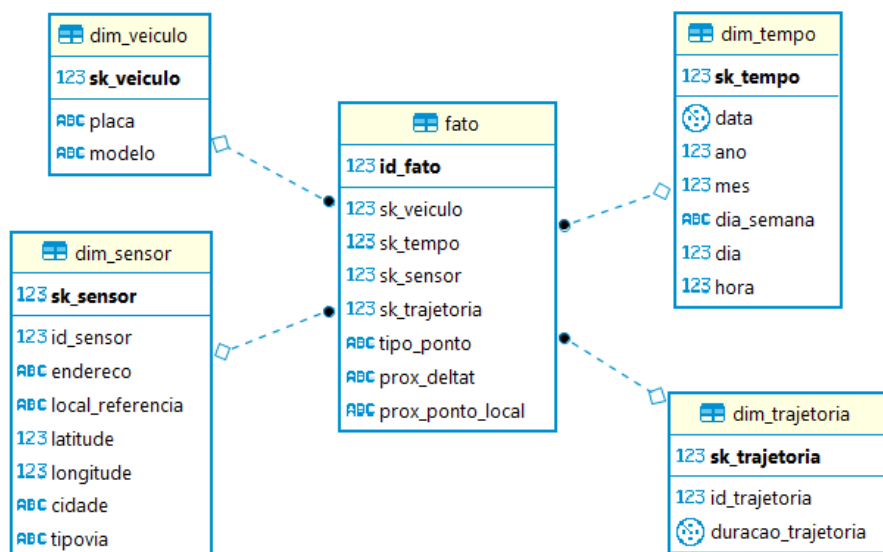
	123 id_registro	123 id_trajetoria	base_data	ABC tipo_ponto	ABC base_cidade	ABC base_endereco	ABC base_placa
1	4	1	2017-09-08 20:49:55	INICIAL	FORTALEZA	AV AUG DOS ANJ X RUA JUL BRA	0000ea8347b3f687106ba2
2	5	1	2017-09-08 20:57:15	FINAL	FORTALEZA	RUA GAL OSO DE PAI, 5841	0000ea8347b3f687106ba2
3	10	2	2017-09-09 16:26:09	INICIAL	FORTALEZA	AV WAS SOA x AV VER PED PAU	000023d2fa5f939eae4788:
4	11	2	2017-09-09 16:27:10	FINAL	FORTALEZA	AV WAS SOA x AV EDS BRA SRS	000023d2fa5f939eae4788:
5	15	3	2017-09-12 15:23:18	INICIAL	FORTALEZA	AV ALB CRA, 1480	000023d2fa5f939eae4788:
6	16	3	2017-09-12 15:26:52	FINAL	FORTALEZA	AV JUS KUB, 3700	000023d2fa5f939eae4788:
7	17	4	2017-09-06 21:18:20	INICIAL	FORTALEZA	CE 040 Km 002 C	000028a7720797adfce726b
8	18	4	2017-09-06 21:19:13	INTERMEDIARIO	FORTALEZA	AV WAS SOA x AV ALM MAX FON	000028a7720797adfce726b
9	19	4	2017-09-06 21:20:34	INTERMEDIARIO	FORTALEZA	CE 040 Km 003 C	000028a7720797adfce726b
10	20	4	2017-09-06 21:22:58	INTERMEDIARIO	FORTALEZA	AV WAS SOA x AV VER PED PAU	000028a7720797adfce726b
11	21	4	2017-09-06 21:24:04	FINAL	FORTALEZA	AV WAS SOA x AV EDS BRA SRS	000028a7720797adfce726b

Fonte: Elaborado pelo autor

4.10 Carga

Depois que os dados foram inseridos na tabela registro_trajetoria, as tabelas dimensões dim_trajetoria, dim_sensor, dim_veiculo e dim_tempo foram criadas, com os comandos 22, 23, 24, 25 e os dados foram inseridos nestas tabelas com os comandos 27, 28, 29, 30 respectivamente. Logo após as tabelas dimensões terem sido povoadas foi efetuada uma carga inicial de dados na tabela fato com o comando 31, onde cada linha armazenada na tabela fato corresponde a um conjunto de identificadores das tabelas dimensões que compõem um registro, com mais três atributos adicionais: tipo_ponto, prox_deltat, prox_ponto_local, o modelo multidimensional construído está definido na Figura 15.

Figura 15 – Modelo Multidimensional Final



Fonte: Elaborado pelo autor

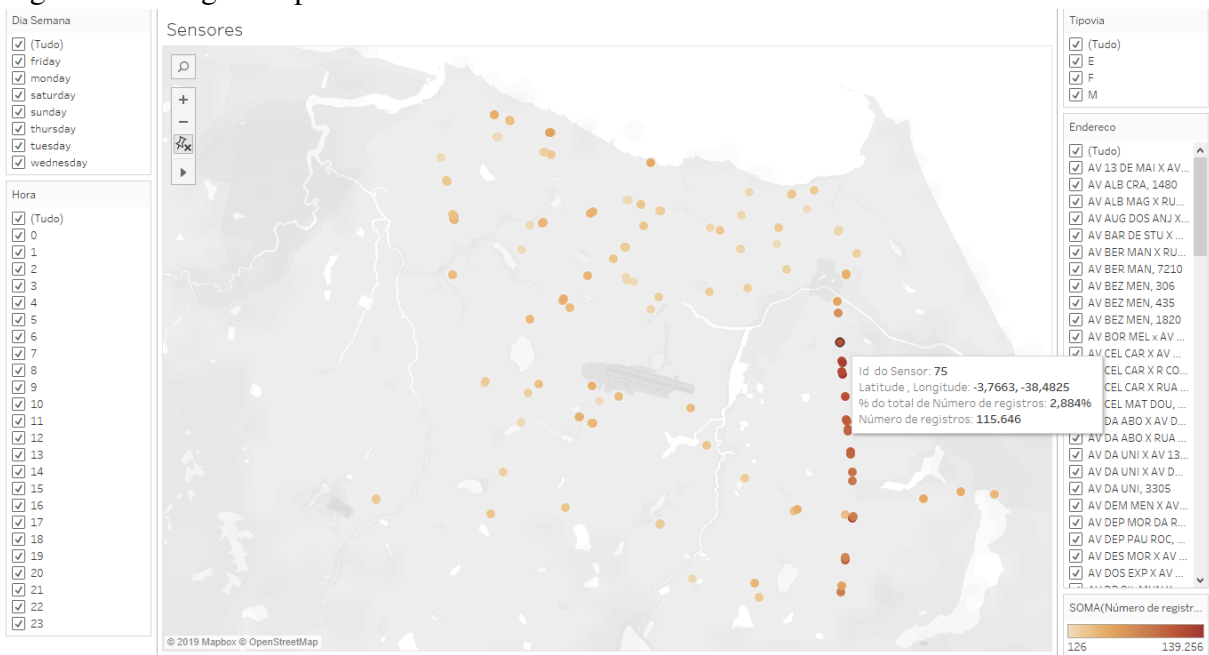
5 RESULTADOS

Neste capítulo são apresentados os resultados obtidos neste projeto. Onde foi utilizada a ferramenta Tableau Desktop para construir as visualizações das informações mostradas a baixo. Tais informações são capazes de responder as perguntas norteadoras específicas.

- Qual a porcentagem de registros por cada sensor instalado, baseado no total?

A Figura 16 mostra os pontos onde cada sensor está instalado, quanto mais escura a cor do ponto significa uma maior quantidade de registros efetuados pelo mesmo. Ao colocar o mouse por cima de um ponto específico é mostrado o identificador do sensor a latitude e longitude de onde está localizado, o número de registros efetuados e porcentagem sobre todos os registros efetuados.

Figura 16 – Registros por sensor

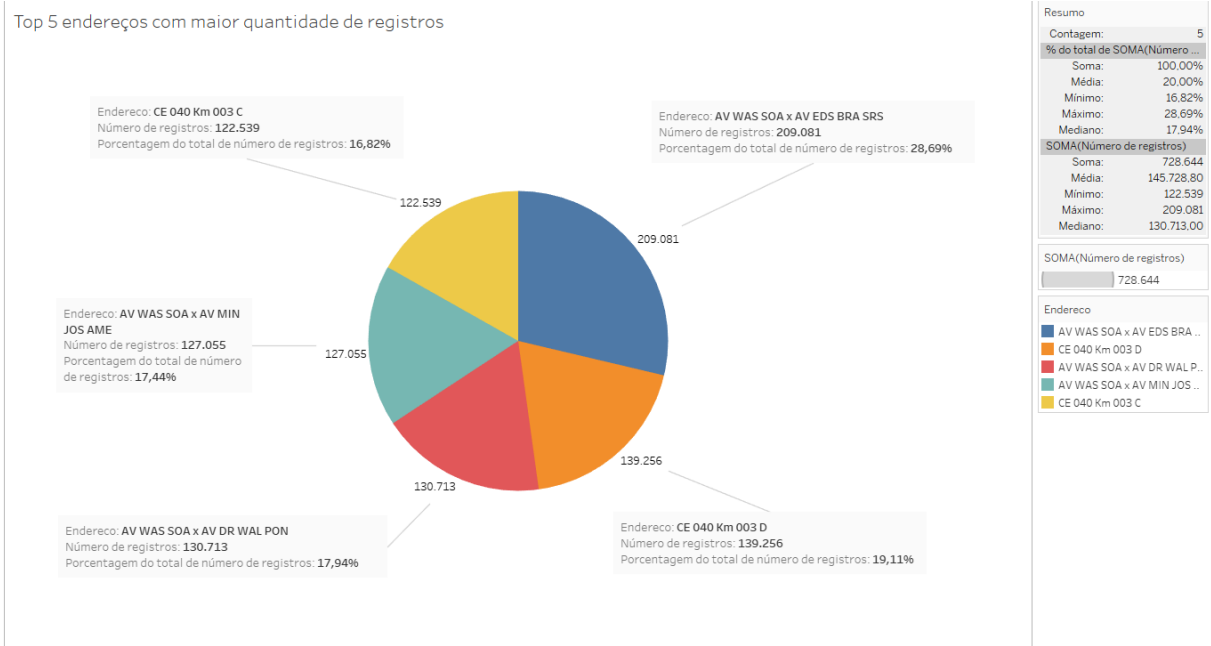


Fonte: Elaborado pelo autor

- Quais são os endereços com maior quantidade de registros?

Na Figura 17 é mostrado um gráfico pizza com os 5 endereços que mais aparecem nos registros, mostrando o nome do endereço, sua quantidade de registros e porcentagem sobre os outros endereços do gráfico.

Figura 17 – Top 5 regiões em quantidade de registros

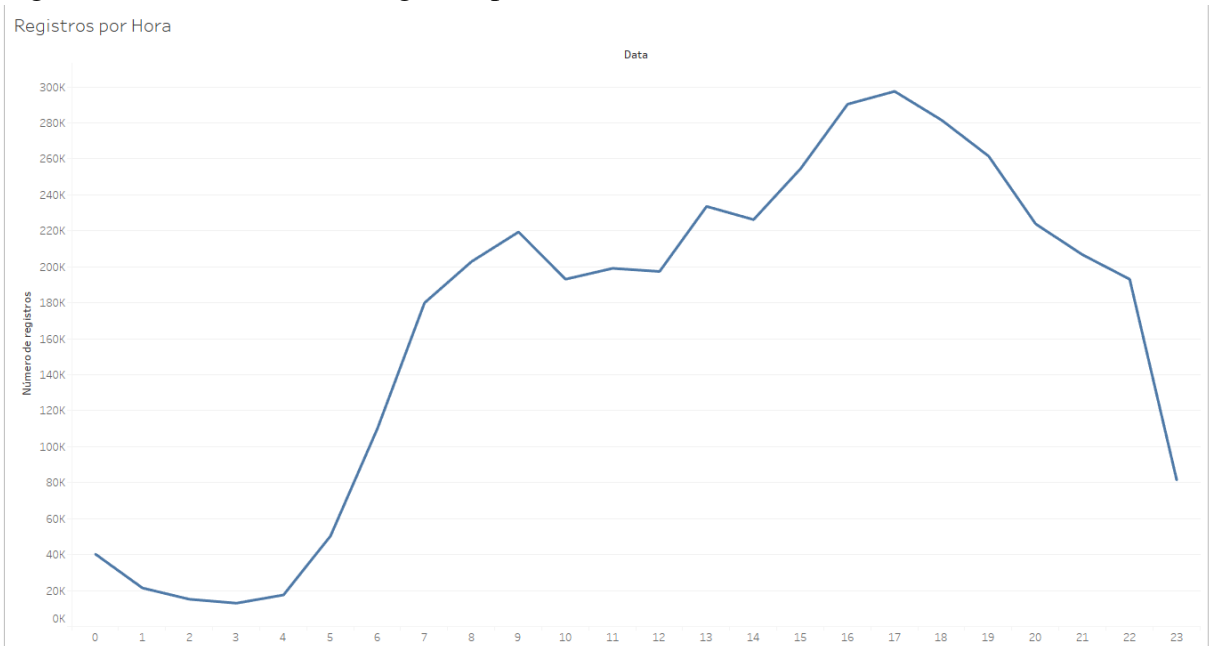


Fonte: Elaborado pelo autor

- Como varia a quantidade de registros pelo tempo, mostrando quais são os horários com maior e menor quantidade de registros?

A Figura 18 está um gráfico de linha mostrando como varia a quantidade de registros por hora, podendo ser identificado qual os horários com menores e maiores quantidade de registros.

Figura 18 – Quantidade de registros por hora

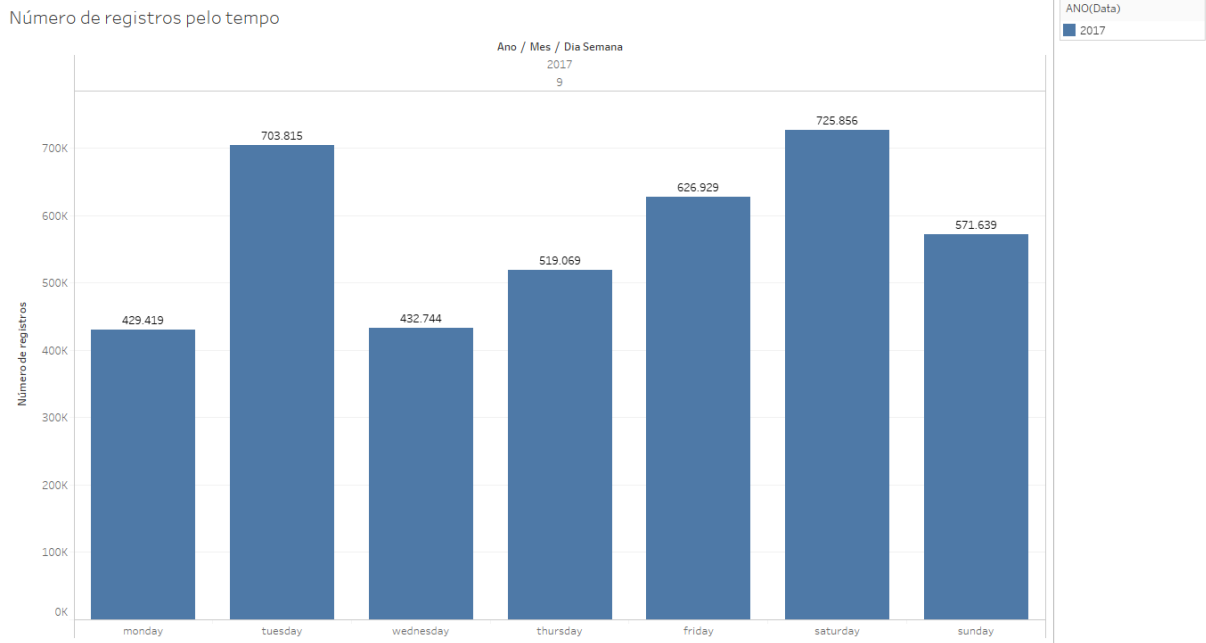


Fonte: Elaborado pelo autor

- Quantos registros no total foram efetuados em um determinado dia e hora?

O gráfico da Figura 19 mostra a quantidade de registros efetuados por dia da semana, assim podendo identificar quais dias geralmente tem menor e maior tráfego de veículos.

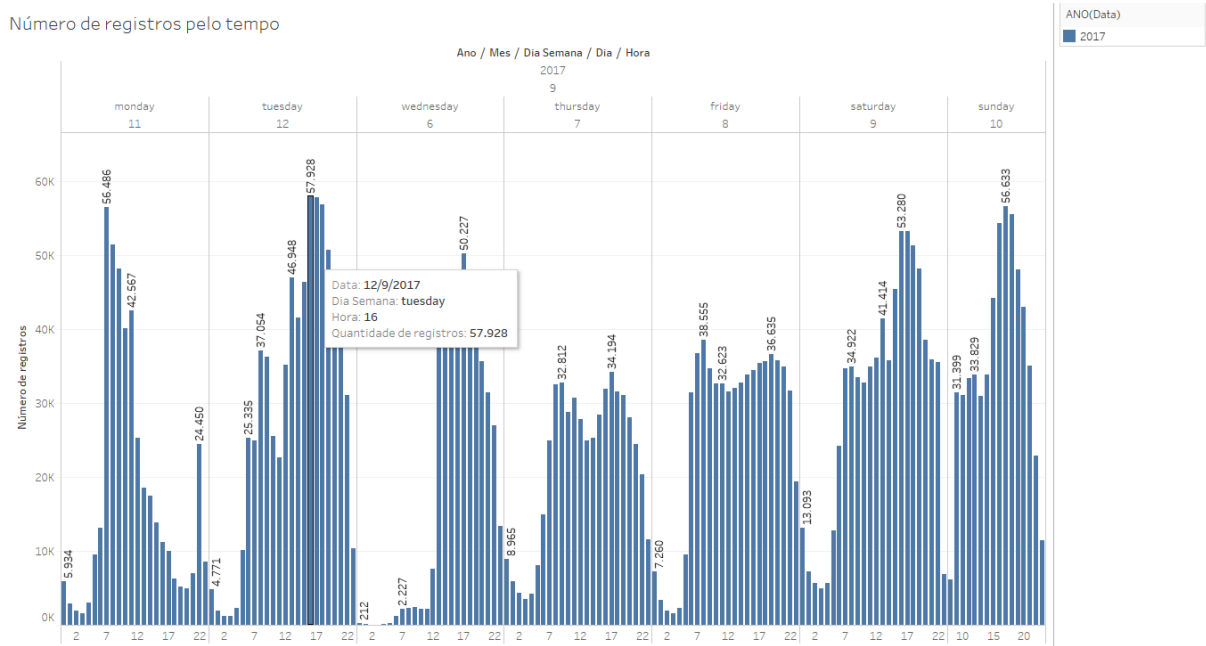
Figura 19 – Quantidade de registros por dia



Fonte: Elaborado pelo autor

Na Figura 20 é mostrado a quantidade de registros efetuados por hora de cada dia, podendo ser vista a diferença entre os horários de cada dia.

Figura 20 – Quantidade de registros por hora do dia

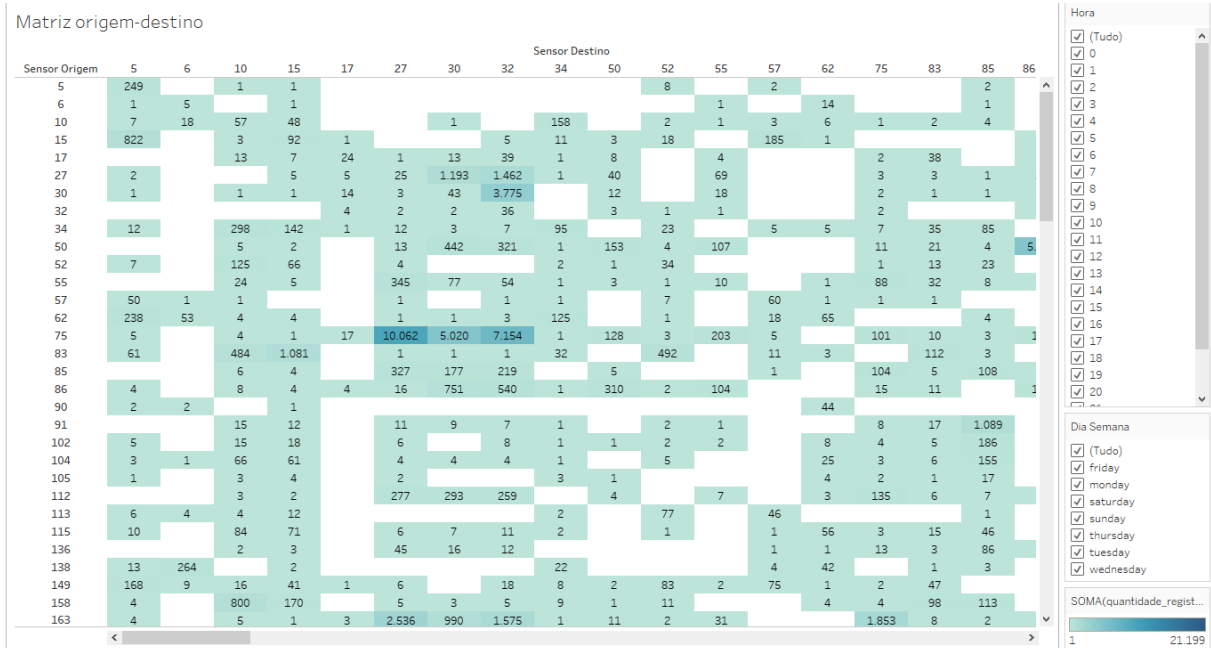


Fonte: Elaborado pelo autor

- Como se dá o fluxo de veículos entre diferentes sensores e em quais dias e horários?

Na Figura 21 é mostrada uma matriz origem-destino, onde cada linha representa um sensor de origem e cada coluna representa um sensor de destino, e as células são as quantidades de trajetórias que começam no sensor origem e terminam no sensor destino. Podendo ser utilizados filtros sobre as horas e os dias da semana.

Figura 21 – Matriz origem-destino



Fonte: Elaborado pelo autor

- Reconstruir as trajetórias feitas e descobrir quanto tempo elas duram.

Na Figura 22 é exibida uma trajetória composta por um conjunto sensores que efetuaram registros de um veículo ordenados pelo tempo. Ao deixar o mouse sobre um ponto é mostrado todas as informações sobre o ponto atual, o tipo do ponto, o tempo que levou para ser efetuado o próximo registro e o tempo total que a trajetória levou para ser percorrida.

Figura 22 – Mapa de trajetórias



Fonte: Elaborado pelo autor

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste artigo, foi proposto um modelo de dados multidimensional para dados de sensores instalados em semáforos e radares de velocidade localizados na cidade de Fortaleza - CE. A partir dos resultados observados no Capítulo 5, podemos concluir que o modelo pôde responder de forma eficiente as perguntas norteadoras destacadas neste trabalho. Mostra-se assim a capacidade de extrair informação útil e que possa auxiliar a tomada de decisão, a partir da utilização de um modelo multidimensional construídos a partir de dados espaço-temporais. O desempenho do modelo em responder às questões propostas e sua usabilidade em uma ferramenta de visualização de dados, tornando a análise de informações mais completa e intuitiva, demonstra que a utilização do modelo proposto neste trabalho para a construção de informação que contribua para a tomada de decisão é válida.

Como trabalho futuro, pode ser proposta a evolução do modelo desenvolvido neste trabalho, com a adição de diferentes granularidades na dimensão espacial como: Distritos, zonas e setores. Também como a possibilidade de se efetuar cargas incrementais de dados sobre o modelo.

REFERÊNCIAS

- ANZANELLO, C. A. **OLAP conceitos e utilização**. Porto Alegre: UFRGS-Instituto de Informática–Universidade Federal do Rio Grande do Sul. 2007.
- ASSOCIAÇÃO NACIONAL DOS DETRANS. **População e Frota total de veículos do estado do Ceará**. 2019. Disponível em: <http://www.and.org.br/detrans/prettyPhoto/21>, Acesso em: 04 dez. 2019.
- BRAZ, F.; ORLANDO, S.; ORSINI, R.; RAFFAETA, A.; RONCATO, A.; SILVESTRI, C. Approximate aggregations in trajectory data warehouses. In: IEEE. **2007 IEEE 23rd International Conference on Data Engineering Workshop**. [S.l.], 2007. p. 536–545.
- BRINKHOFF, T. A framework for generating network-based moving objects. **GeoInformatica**, Springer, v. 6, n. 2, p. 153–180,[S.l.], 2002.
- CAMPOS, M. L. M.; BORGES, V. J. d. A. S. Diretrizes para a modelagem incremental de data marts. In: **SBBD**. [S.l.: s.n.], 2002. p. 110–120.
- DIARIO DO NORDESTE. **Frota de veículos de Fortaleza é a maior da região Nordeste**. 2016. Disponível em: <https://diariodonordeste.verdesmares.com.br/editorias/metro/frota-de-veiculos-de-fortaleza-e-a-maior-da-regiao-nordeste-1.1535497>, Acesso em: 04 dez. 2019.
- ELMASRI, R.; NAVATHE, S. **Fundamentals of database systems**. [S.l.]: Addison-Wesley Publishing Company, 2010.
- EMPRESÔMETRO, I. **REAL FROTA CIRCULANTE NO BRASIL É DE 65,8 MILHÕES DE VEÍCULOS, INDICA ESTUDO**. 2018. Disponível em: <https://ibpt.com.br/noticia/2640>, Acesso em: 04 dez. 2019.
- GONÇALVES, M. **Extração de dados para data warehouse**: Rio de Janeiro/RJ: Axcell Books. 2003.
- INMON, W. H. **Como construir o data warehouse**. [S.l.]: Campus, 1997.
- KIMBALL, R.; CASERTA, J. **The data warehouse ETL toolkit**: practical techniques for extracting, cleaning, conforming, and delivering data. [S.l.]: Wiley, 2004.
- KIMBALL, R.; ROSS, M. **The data warehouse toolkit**: the definitive guide to dimensional modeling. [S.l.]: John Wiley & Sons, 2013.
- LEONARDI, L.; MARKETOS, G.; FRENTZOS, E.; GIATRAKOS, N.; ORLANDO, S.; PELEKIS, N.; RAFFAETA, A.; RONCATO, A.; SILVESTRI, C.; THEODORIDIS, Y. **T-Warehouse**: visual olap analysis on trajectory data. In: Springer. [S.l.], 2010. [S. l.: s. n.], p. 1141–1144.
- MACHADO, F. N. R. **Projeto de Data Warehouse**: uma visão multidimensional. [S.l.]: Ed. Érica, 2000.
- PONNIAH, P. **Data warehousing fundamentals for IT professionals**. [S.l.]: John Wiley & Sons, 2011.

WAGNER, R.; MACEDO, J. A. F. de; RAFFAETÀ, A.; RENSO, C.; RONCATO, A.; TRASARTI, R. Mob-warehouse: a semantic approach for mobility analysis with a trajectory data warehouse. In: Springer. **International Conference on Conceptual Modeling**. [S.l.], 2013. p. 127–136.

APÊNDICE A – CÓDIGOS SQL UTILIZADOS PARA PRODUÇÃO DO PRIMEIRO MODELO MULTIDIMENSIONAL

Código-fonte 1 – Criando tabela base_extracao

```
1 CREATE TABLE base_extracao(  
2     base_data timestamp,  
3     base_cidade varchar(255),  
4     base_endereco varchar(255),  
5     base_branco varchar(255),  
6     base_placa varchar(255),  
7     base_modelo varchar(255),  
8     base_cor varchar(255),  
9     base_estado varchar(255),  
10    base_dezena varchar(255),  
11    base_unidade varchar(255),  
12    base_situacao varchar(255),  
13    base_ais varchar(255),  
14    base_cidadex varchar(255),  
15    base_tipovia varchar(255),  
16    base_local_referencia varchar(255));
```

Código-fonte 2 – Criando tabela sensores_extracao

```
1 CREATE TABLE sensores_extracao(  
2     sensor_id int,  
3     sensor_id_2 int,  
4     sensor_endereco varchar(255),  
5     sensor_cidade varchar(255),  
6     sensor_tipovia varchar(255),  
7     sensor_local_referencia varchar(255),
```

```

8     sensor_latitude float,
9     sensor_longitude float);

```

Código-fonte 3 – Populando tabela base_tratamento

```

1 INSERT INTO base_tratamento ( base_data, base_cidade,
2     base_endereco, base_placa, base_modelo, base_cor,
3     base_tipovia, base_local_referencia)
4 SELECT DISTINCT base_data, base_cidade, base_endereco,
5     base_placa, base_modelo, base_cor, base_tipovia,
6     base_local_referencia
7 FROM base_extracao WHERE base_cidade = "FORTALEZA"
8 AND base_data::timestamp < "2017-09-02 00:00:00"::timestamp
9 ORDER BY base_data;

```

Código-fonte 4 – Populando tabela sensores_tratamento

```

1 INSERT INTO sensores_tratamento (sensor_id, sensor_endereco
2     ,
3     sensor_cidade, sensor_tipovia, sensor_local_referencia,
4     sensor_latitude, sensor_longitude)
5 SELECT DISTINCT sensor_id, sensor_endereco, sensor_cidade,
6     sensor_tipovia, sensor_local_referencia,
7     sensor_latitude, sensor_longitude
8 FROM sensores_extracao WHERE sensor_cidade = "FORTALEZA";

```

Código-fonte 5 – Populando tabela tabela_master

```

1 INSERT INTO tabela_master ( base_data, base_cidade,
2     base_endereco, base_placa, base_modelo, base_tipovia,

```

```

3     base_local_referencia, sensor_id, sensor_endereco,
4     sensor_cidade, sensor_tipovia, sensor_local_referencia,
5     sensor_latitude, sensor_longitude)
6 SELECT distinct * FROM base_tratamento b,
7     sensores_tratamento s
8 WHERE b.base_endereco = s.sensor_endereco
9 AND b.base_local_referencia = s.sensor_local_referencia
10 ORDER BY b.base_data;

```

Código-fonte 6 – Populando tabela dimensão dim_sensor

```

1 INSERT INTO dim_sensor(id_sensor, endereco,
2     local_referencia, latitude, longitude, cidade, tipovia)
3 SELECT distinct b.sensor_id, b.sensor_endereco,
4     b.sensor_local_referencia, b.sensor_latitude,
5     b.sensor_longitude, b.base_cidade, s.sensor_tipovia
6 FROM tabela_master b, sensores_tratamento s
7 WHERE s.sensor_endereco = b.base_endereco
8 AND s.sensor_local_referencia = b.base_local_referencia
9 AND s.sensor_cidade = b.base_cidade ORDER BY b.sensor_id;

```

Código-fonte 7 – Populando tabela dimensão dim_veiculo

```

1 INSERT INTO dim_veiculo(placa, modelo)
2 SELECT distinct b.base_placa, b.base_modelo
3 FROM tabela_master b, sensores_tratamento s
4 WHERE b.base_placa != ""
5 AND s.sensor_endereco = b.base_endereco
6 AND s.sensor_local_referencia = b.base_local_referencia
7 AND s.sensor_cidade = b.base_cidade ORDER BY b.base_placa;

```

Código-fonte 8 – Populando tabela dimensão dim_tempo

```

1 INSERT INTO dim_tempo("data", ano , mes , dia, hora)
2 SELECT distinct b.base_data,
3     EXTRACT(YEAR FROM b.base_data) AS ANO,
4     EXTRACT(MONTH FROM b.base_data) AS MES,
5     EXTRACT(DAY FROM b.base_data) AS DIA,
6     EXTRACT(HOUR FROM b.base_data) AS HORA
7 FROM tabela_master b WHERE b.base_placa != ""
8 AND b.sensor_endereco = b.base_endereco
9 AND b.sensor_local_referencia = b.base_local_referencia
10 AND b.sensor_cidade = b.base_cidade order by b.base_data;

```

Código-fonte 9 – Criando tabela fato

```

1 CREATE TABLE fato(
2     id_fato serial NOT NULL PRIMARY KEY,
3     sk_veiculo INT,
4     sk_tempo INT,
5     sk_sensor INT,
6 FOREIGN KEY (sk_veiculo)REFERENCES dim_veiculo(sk_veiculo),
7 FOREIGN KEY (sk_tempo)REFERENCES dim_tempo(sk_tempo),
8 FOREIGN KEY (sk_sensor)REFERENCES dim_sensor(sk_sensor));

```

Código-fonte 10 – Populando tabela fato

```

1 INSERT INTO fato(sk_veiculo, sk_tempo, sk_sensor)
2 SELECT distinct v.sk_veiculo, t.sk_tempo, s.sk_sensor
3 FROM dim_veiculo v,dim_tempo t,dim_sensor s,tabela_master b
4 WHERE b.sensor_id = s.id_sensor AND b.base_data = t."data"

```

```
5 AND b.base_placa = v.placa;
```

APÊNDICE B – CÓDIGOS SQL UTILIZADOS PARA A ETAPA DE REFINAMENTO DO MODELO

Código-fonte 11 – Criando tabela base_extração

```
1 CREATE TABLE "Staging Area".registro_extracao(  
2     base_data timestamp,  
3     base_cidade varchar(255),  
4     base_endereco varchar(255),  
5     base_branco varchar(255),  
6     base_placa varchar(255),  
7     base_modelo varchar(255),  
8     base_cor varchar(255),  
9     base_estado varchar(255),  
10    base_dezena varchar(255),  
11    base_unidade varchar(255),  
12    base_situacao varchar(255),  
13    base_ais varchar(255),  
14    base_cidadex varchar(255),  
15    base_tipovia varchar(255),  
16    base_local_referencia varchar(255));
```

Código-fonte 12 – Criando tabela sensor_extracao

```
1 CREATE TABLE "Staging Area".sensor_extracao(  
2     sensor_id int,  
3     sensor_id_2 int,  
4     sensor_endereco varchar(255),  
5     sensor_cidade varchar(255),  
6     sensor_tipovia varchar(255),  
7     sensor_local_referencia varchar(255),
```

```
8     sensor_latitude float ,  
9     sensor_longitude float);
```

Código-fonte 13 – Criando Schema Staging Area

```
1 CREATE SCHEMA "Staging Area";
```

Código-fonte 14 – Criando Schema Data Warehouse

```
1 CREATE SCHEMA "Data Warehouse";
```

Código-fonte 15 – Criando tabela registro_trajetoria

```
1 CREATE TABLE "Staging Area".registro_trajetoria(  
2     id_registro serial NOT null primary key,  
3     id_trajetoria BIGSERIAL,  
4     base_data "timestamp",  
5     tipo_ponto varchar,  
6     base_cidade varchar(255),  
7     base_endereco varchar(255),  
8     base_placa varchar(255),  
9     base_modelo varchar(255),  
10    base_tipovia varchar(255),  
11    base_local_referencia varchar(255),  
12    sensor_id int,  
13    sensor_endereco varchar(255),  
14    sensor_cidade varchar(255),  
15    sensor_tipovia varchar(255),  
16    sensor_local_referencia varchar(255),  
17    sensor_latitude float,
```

```

18 sensor_longitude float,
19 pre_data "timestamp",
20 pre_deltat interval,
21 prox_data "timestamp",
22 prox_deltaT interval);

```

Código-fonte 16 – Criando função que verifica os dados antes de inseri-los na tabela registro_trajetoria

```

1 CREATE OR REPLACE FUNCTION "Staging Area".limparRegisgtros
  ()
2 RETURNS trigger AS $$
3 BEGIN
4     IF new.pre_deltat::interval = "00:00:00"::interval
5     or new.prox_deltat::interval = "00:00:00"::interval
6     THEN
7         RETURN null;
8     ELSE
9         IF new.pre_deltat::interval > "00:20:00"::interval
10        THEN
11            new.pre_data = null;
12        END IF;
13        IF new.prox_deltaT ::interval > "00:20:00"::interval
14        THEN
15            new.prox_data = null;
16        END IF;
17        IF new.pre_data isnull and new.prox_data isnull
18        THEN
19            RETURN null;
20        END IF;
21        RETURN new;
22    END IF;

```



```

19 END;
20 $$ LANGUAGE plpgsql;

```

Código-fonte 17 – Criando *trigger* que ativa a função 16

```

1 CREATE TRIGGER limparRegistros_trigger BEFORE insert
2 ON "Staging Area".registro_trajetoria
3 FOR EACH ROW
4 EXECUTE PROCEDURE "Staging Area".limparRegisgtros();

```

Código-fonte 18 – Criando função que adiciona o atributo identificador das trajetórias e os tipos dos pontos

```

1 CREATE OR REPLACE FUNCTION "Staging Area".add_idTrajetoria
2   ()
3 RETURNS trigger AS $$
4 DECLARE
5   interator int;
6 BEGIN
7   SELECT incremental INTO interator from "Staging Area".
8     temp_tb;
9   if new.pre_data isnull and new.prox_data notnull
10    then
11     UPDATE "Staging Area".registro_trajetoria SET
12       id_trajetoria = interator, tipo_ponto = "INICIAL
13     " WHERE id_registro = NEW.id_registro;
14   return new;
15 ELSIF new.pre_data notnull and new.prox_data notnull
16 then
17   UPDATE "Staging Area".registro_trajetoria SET
18     id_trajetoria = interator, tipo_ponto = "

```

```

INTERMEDIARIO"
12 WHERE id_registro = NEW.id_registro;
13     return new;
14 ELSIF new.pre_data notnull and new.prox_data isnull
    then
15 UPDATE "Staging Area".registro_trajetoria SET
    id_trajetoria = interator, tipo_ponto = "FINAL"
16 WHERE id_registro = NEW.id_registro;
17     update "Staging Area".temp_tb set incremental =
    interator + 1;
18     return new;
19 END IF;
20 END;
21 $$ LANGUAGE plpgsql;

```

Código-fonte 19 – Criando *trigger* que ativa a função 18

```

1 create TRIGGER idTrajetoria_trigger after insert ON
2 "Staging Area".registro_trajetoria
3 FOR EACH row EXECUTE PROCEDURE "Staging Area".
    add_idTrajetoria();

```

Código-fonte 20 – Criando tabela temp_tb

```

1 create table "Staging Area".temp_tb(
2     incremental bigserial
3 );
4 insert into "Staging Area".temp_tb values(1);

```

Código-fonte 21 – Poulando tabela registro_trajetoria

```
1 insert into "Staging Area".registro_trajetoria(  
2     base_data ,  
3     base_cidade ,  
4     base_endereco ,  
5     base_placa ,  
6     base_modelo ,  
7     base_tipovia ,  
8     base_local_referencia ,  
9     sensor_id ,  
10    sensor_endereco ,  
11    sensor_cidade ,  
12    sensor_tipovia ,  
13    sensor_local_referencia ,  
14    sensor_latitude ,  
15    sensor_longitude ,  
16    pre_data ,  
17    pre_deltat ,  
18    prox_data ,  
19    prox_deltaT  
20 )SELECT distinct  
21     base_data ,  
22     base_cidade ,  
23     base_endereco ,  
24     base_placa ,  
25     base_modelo ,  
26     base_tipovia ,  
27     base_local_referencia ,  
28     sensor_id, sensor_endereco , sensor_cidade ,  
29     sensor_tipovia , sensor_local_referencia , sensor_latitude ,  
30     sensor_longitude ,  
31     LAG (b.base_data , 1) OVER (PARTITION BY b.base_placa
```

```

ORDER BY b.base_data)
32 AS pre_data, b.base_data - LAG (b.base_data, 1)
33 OVER ( PARTITION BY b.base_placa ORDER BY b.base_data)
34 AS pre_deltat,
35 LEAD(b.base_data, 1) OVER(PARTITION BY b.base_placa
ORDER BY b.base_data)
36 AS prox_data, LEAD (b.base_data, 1) OVER ( PARTITION BY b
.base_placa ORDER BY b.base_data) - b.base_data
37 AS prox_deltaT
38 FROM "Staging Area".registro_extracao b inner join "Staging
Area".sensor_extracao s on b.base_cidade = s.
sensor_cidade and
39 b.base_endereco = s.sensor_endereco and b.
base_local_referencia = s.sensor_local_referencia
40 WHERE s.sensor_cidade = "FORTALEZA" AND b.base_data between
"2017-09-06 00:00:00" and "2017-09-12 23:59:59"
41 ORDER BY base_placa, base_data;

```

Código-fonte 22 – Criando tabela dim_trajetoria

```

1 create table "Data Warehouse".dim_trajetoria(
2     sk_trajetoria serial not null primary key,
3     id_trajetoria int,
4     duracao_trajetoria interval);

```

Código-fonte 23 – Criando tabela dim_sensor

```

1 CREATE TABLE "Data Warehouse".dim_sensor (
2     sk_sensor serial NOT NULL PRIMARY KEY,
3     id_sensor int,

```

```
4   endereco varchar(255),
5   local_referencia varchar(255),
6   latitude double precision,
7   longitude double precision,
8   cidade varchar(255),
9   tipovia varchar(5));
```

Código-fonte 24 – Criando tabela dim_veiculo

```
1 CREATE TABLE "Data Warehouse".dim_veiculo(
2   sk_veiculo serial NOT NULL PRIMARY KEY,
3   placa varchar(255),
4   modelo varchar(255));
```

Código-fonte 25 – Criando tabela dim_tempo

```
1 CREATE TABLE "Data Warehouse".dim_tempo(
2   sk_tempo serial NOT NULL PRIMARY KEY,
3   "data" TIMESTAMP,
4   ano DOUBLE PRECISION,
5   mes DOUBLE PRECISION,
6   dia_semana VARCHAR(15),
7   dia DOUBLE PRECISION,
8   hora DOUBLE PRECISION);
```

Código-fonte 26 – Criando tabela fato

```
1 CREATE TABLE "Data Warehouse".fato(
2   id_fato serial NOT NULL PRIMARY KEY,
3   sk_veiculo INT,
```

```

4   sk_tempo INT,
5   sk_sensor INT,
6   sk_trajetoria INT,
7   tipo_ponto varchar,
8   prox_deltat TEXT,
9   prox_ponto_local varchar,
10  FOREIGN KEY (sk_veiculo) REFERENCES "Data Warehouse".
    dim_veiculo (sk_veiculo),
11  FOREIGN KEY (sk_tempo) REFERENCES "Data Warehouse".
    dim_tempo (sk_tempo),
12  FOREIGN KEY (sk_sensor) REFERENCES "Data Warehouse".
    dim_sensor (sk_sensor),
13  FOREIGN KEY (sk_trajetoria) REFERENCES "Data Warehouse".
    dim_trajetoria (sk_trajetoria));

```

Código-fonte 27 – Populando tabela dim_trajetoria

```

1  insert into "Data Warehouse".dim_trajetoria(
2    id_trajetoria,
3    duracao_trajetoria
4  ) SELECT INICIO.id_trajetoria , (FIM.time1 - INICIO.time2)
    duracao_trajetoria FROM
5    (select id_trajetoria, base_data as time1
6    from "Staging Area".registro_trajetoria
7    WHERE tipo_ponto = "FINAL"
8    group by id_trajetoria, base_data
9    ORDER BY id_trajetoria) FIM,
10   (select id_trajetoria, base_data as time2
11   from "Staging Area".registro_trajetoria
12   WHERE tipo_ponto = "INICIAL"
13   group by id_trajetoria, base_data

```

```
14 ORDER BY id_trajetoria) INICIO
15 WHERE FIM.id_trajetoria = INICIO.id_trajetoria;
```

Código-fonte 28 – Populando tabela dim_sensor

```
1 INSERT INTO "Data Warehouse".dim_sensor (
2     id_sensor ,
3     endereco ,
4     local_referencia ,
5     latitude ,
6     longitude ,
7     cidade ,
8     tipovia)
9 SELECT distinct b.sensor_id, b.sensor_endereco, b.
10     sensor_local_referencia ,
11 b.sensor_latitude, b.sensor_longitude, b.base_cidade, b.
12     sensor_tipovia
13 from "Staging Area".registro_trajetoria b ORDER BY b.
14     sensor_id;
```

Código-fonte 29 – Populando tabela dim_veiculo

```
1 INSERT INTO "Data Warehouse".dim_veiculo(
2     placa ,
3     modelo )
4 SELECT distinct b.base_placa, b.base_modelo
5 FROM "Staging Area".registro_trajetoria b WHERE b.
6     base_placa != ""
7 ORDER BY b.base_placa;
```

Código-fonte 30 – Populando tabela dim_tempo

```

1  INSERT INTO "Data Warehouse".dim_tempo(
2     "data" ,
3     ano ,
4     mes ,
5     dia_semana ,
6     dia ,
7     hora
8 )SELECT distinct b.base_data ,
9  EXTRACT(YEAR FROM b.base_data) AS ANO ,
10 EXTRACT(MONTH FROM b.base_data) AS MES ,
11 to_char(b.base_data, "day") as DIA_SEMANA ,
12 EXTRACT(DAY FROM b.base_data) AS DIA ,
13 EXTRACT(HOUR FROM b.base_data) AS HORA
14 FROM "Staging Area".registro_trajetoria b WHERE b.
15     base_placa != ""
16 AND b.sensor_endereco = b.base_endereco
17 AND b.sensor_local_referencia = b.base_local_referencia
18 AND b.sensor_cidade = b.base_cidade ORDER BY b.base_data;

```

Código-fonte 31 – Populando tabela fato

```

1  INSERT INTO "Data Warehouse".fato(
2     sk_veiculo ,
3     sk_tempo ,
4     sk_sensor ,
5     sk_trajetoria ,
6     tipo_ponto ,
7     prox_deltat ,
8     prox_ponto_local
9 )SELECT distinct v.sk_veiculo , t.sk_tempo , s.sk_sensor ,

```



```

    traj.sk_trajetoria,
10 a.tipo_ponto, case when a.prox_deltat > "00:20:00" then "
    00:00:00"
11 else a.prox_deltat end as prox_deltat,
12 case when a.prox_data notnull
13 then b.base_endereco else "Fim" end as prox_ponto
14 from "Staging Area".registro_trajetoria a inner join "
    Staging Area".registro_trajetoria b
15 on a.id_trajetoria = b.id_trajetoria, "Data Warehouse".
    dim_veiculo v, "Data Warehouse".dim_tempo t, "Data
    Warehouse".dim_sensor s, "Data Warehouse".dim_trajetoria
    traj
16 WHERE (a.prox_data = b.base_data or a.prox_data isnull)
17 and a.sensor_id = s.id_sensor
18 AND a.base_data = t."data"
19 AND a.base_placa = v.placa
20 and a.id_trajetoria = traj.id_trajetoria;

```

Código-fonte 32 – Construindo matriz origem destino

```

1 select f1.id_fato as id_fato, inicio.id_sensor as
    id_sensor_inicio, sfim.id_sensor as id_sensor_fim,
2 count(distinct dtraj.id_trajetoria) as quantidade_registros
3 from "Data Warehouse".dim_sensor inicio, "Data Warehouse".
    dim_sensor sfim, "Data Warehouse".dim_trajetoria dtraj,
4 "Data Warehouse".fato f1, "Data Warehouse".fato f2 WHERE f1
    .sk_sensor = inicio.sk_sensor and f1.tipo_ponto = "
    INICIAL"
5 and f2.sk_sensor = sfim.sk_sensor and f2.tipo_ponto = "
    FINAL"
6 and f1.sk_trajetoria = dtraj.sk_trajetoria

```

```
7 and f2.sk_trajetoria = dtraj.sk_trajetoria
8 group by f1.id_fato, sinicio.id_sensor, sfim.id_sensor
   ORDER BY f1.id_fato, sinicio.id_sensor, sfim.id_sensor;
```