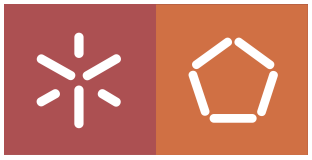

Universidade do Minho
Escola de Engenharia

Análise de Dados Clínicos

COVID-19

Mestrado Integrado em Engenharia Biomédica

Informática Médica

O Processo Clínico Eletrónico

2º semestre

2020/2021

Autores:

Id9248 Alexandre Rafael Machado Oliveira

A83624 João Miguel da Silva Alves

A84480 Paulo Jorge Alves

Docente:

António Abelha

Braga

1 de junho 2021

Resumo

A manipulação e exploração de dados alcançou um patamar de excelência devido às ferramentas de processamento analítico que existem atualmente e devido às diversas plataformas de visualização de dados, na área de *Business Intelligence*.

Por este motivo, foi desenvolvido um *Data Warehouse*, onde se efetuou um modelo lógico, o qual foi carregado com dados provenientes de uma base de dados do *MongoDB*, relacionados com sintomas de Covid-19 de diversos pacientes. Todo o processo de ETL foi devidamente realizado, de forma a todos os dados estarem em concordância com o modelo implementado no *MySQL*.

Por fim, com recurso à plataforma *Tableau*, foram elaborados diversos gráficos, os quais permitem analisar e relacionar os dados, possibilitando uma visualização dos mesmos num formato mais apelativo e intuitivo.

Keywords: Business Intelligence; Data Warehouse; MongoDB; Covid-19; ETL; Tableau.

Índice

Índice de Figuras

1 Introdução

O presente projeto enquadra-se na unidade curricular de Processo Clínico Eletrónio, na qual foi proposta o desenvolvimento de um sistema de *Data Warehousing*, bem como um sistema de *Business Intelligence* de suporte à decisão.

O dataset utilizado no trabalho apresenta dados relativos a vários pacientes com diferentes sintomas de Covid-19. Este dataset é de extrema importância, nomeadamente na atualidade, à custa da pandemia que se vive nos dias de hoje.

Para solucionar da melhor forma o desafio apresentado, recorre-se aos conceitos de extração, transformação e carregamento, abreviados pela sigla, ETL. Deste modo, após a extração e transformação do dataset escolhido, torna-se fundamental proceder ao carregamento dos dados para uma base de dados de suporte à decisão, que é mantida separadamente da base operacional da organização, um *Data Warehouse*. A base de dados escolhida, inicialmente, foi o *MongoDB* para armazenar os dados em csv. De seguida, estes dados foram exportados do *MongoDB* para o *MySQL*, onde se tornou necessário implementar um modelo lógico e carregar os dados provenientes do *MongoDB* para efetuar o povoamento das tabelas.

Relativamente aos processos, estruturas e tecnologias que transformam uma grande quantidade de dados brutos em informação útil para tomadas de decisões estratégicas, usou-se o *Tableau* devido às inúmeras vantagens subjacentes ao mesmo. Esta tecnologia permite a visualização de toda a informação num formato mais apelativo e intuitivo.

A fim de garantir o cumprimento de todos os requisitos propostos, procedeu-se à implementação de um Sistema de Base de Dados Multidimensional completo que seja capaz de analisar e tomar decisões de apoio clínico. Assim, segue-se a explicação detalhada de todo o processo de elaboração do referido projeto.

O presente trabalho está dividido em 4 grandes capítulos. No primeiro capítulo está presente um Estado da Arte, onde se refere os principais temas que estão relacionados com o projeto e os respetivos dados manipulados. No segundo, menciona-se uma breve contextualização do trabalho, constatando como se fez o planeamento e a implementação do projeto. De seguida, descreve-se todo o processo de implementação do *Data Warehouse* e o processo de ETL para se conseguir carregar os dados no mesmo. Por fim, apresentam-se todos os gráficos realizados, utilizando a plataforma *Tableau*, para uma melhor visualização e compreensão dos mesmos.

2 Estado da Arte

2.1 COVID-19

A Covid-19 é uma doença respiratória causada pelo vírus *SARS-CoV-2*, pertencente à família dos coronavírus. É uma infeção que se inicia com um quadro semelhante ao da gripe, no entanto, pode agravar-se, podendo levar à morte.

Os primeiros casos de Covid-19 surgiram na cidade de Wuhan, na China, em dezembro de 2019, apresentando quadros de pneumonia de causa desconhecida. Como a doença apresenta uma grande transmissibilidade, começaram, de imediato, a surgir casos em outros países de todo o mundo. No dia 11 de março de 2020, a Organização Mundial de Saúde (OMS) decretou estado de pandemia.

A Covid-19 apresenta um período de incubação (período entre o contágio e o surgimento dos sintomas) de cerca de 14 dias. A maioria das pessoas infetadas desenvolve a doença com sintomas ligeiros a moderados e recupera sem necessidade de hospitalização. Os sintomas mais comuns são: febre, tosse seca, cansaço, perda de paladar ou olfato, etc., no entanto, algumas pessoas podem apresentar agravamento da doença, desenvolvendo dificuldade respiratória e podendo, inclusive, morrer. As pessoas idosas e indivíduos que apresentam certos problemas de saúde, como hipertensão, problemas cardíacos e diabetes, estão mais propensas ao agravamento da doença.

Esta doença é transmitida, principalmente, de uma pessoa para outra por meio das gotículas respiratórias. Entre as medidas para prevenir o contágio e evitar a disseminação da doença, destaca-se a importância da higienização/desinfeção frequente das mãos e o evitar aglomerações. É importante destacar que, em alguns casos, um indivíduo contaminado pode transmitir a doença, mesmo antes de apresentar sintomas [1].

2.2 MongoDB

O *MongoDB* é uma base de dados orientada a documentos, também chamada de base de dados NoSQL (*Not Only SQL*), de código aberto e escrita na linguagem C++.

Bases de dados *NoSQL* têm como características conter todas as informações importantes num único documento, ser livre de esquemas, ao contrário do *SQL*, onde tudo é representado usando uma abordagem bidimensional (tabelas representadas através de duas dimensões: linhas e colunas), possuir identificadores únicos universais (UUID), possibilitar a consulta de documentos através de métodos avançados de agrupamento e filtragem (MapReduce) e também permitir redundância e inconsistência.

Com o *MongoDB* existe um melhor desempenho, visto que, com uma única query (mais simples, visto que não existem joins), é retornado tudo o que é pretendido sobre o documento. As bases de dados *NoSQL* apresentam vantagens sobre as restantes quando é requerido escalabilidade, flexibilidade, manipulação de quantidade massiva de dados e bom desempenho.

Outra vantagem do uso deste tipo de bases de dados é o escalonamento com *Sharding*, que é muito bem implementado no *MongoDB*. *Sharding* é um método para distribuir dados em várias máquinas. O *MongoDB* usa este mecanismo para suportar implantações com conjuntos de dados muito grandes e operações de alto rendimento. Portanto, quanto mais *Shards* maior será o armazenamento e o desempenho. Pelo contrário, bases de dados relacionais muito utilizadas, como o *MySQL*, não suportam este tipo de mecanismos [2].

2.3 Data Warehouse

A ideia principal de um sistema de *Data Warehouse* (DW), ilustrado na **figura 1**, consiste em agregar informação proveniente de uma ou mais bases de dados, ou de outras fontes, para posteriormente a tratar, formatar e consolidar numa única estrutura de dados.

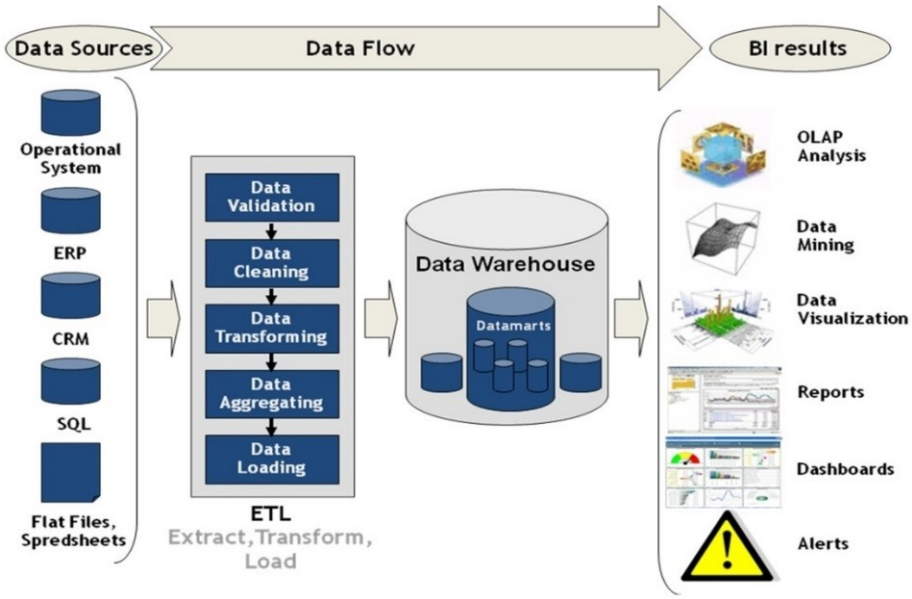

Figura 1: *Data Warehouse* [3]

Um sistema DW está associado a bases de dados com um grande volume de dados devido quer ao volume proveniente das fontes heterogéneas, quer da baixa normalização habitualmente utilizada. A estrutura de dados do DW é desenvolvida de forma a facilitar a análise desses dados. Após ser armazenada, esta informação fica disponível no DW para consultas que visam ajudar na tomada de decisão.

Para a construção de um DW são necessários diferentes passos, principalmente ao nível da extração e processamento de dados.

ETL (*Extract, Transform, Load*) - **figura 2** - é um processo que extrai e compila os dados de diferentes sistemas de origem, transforma-os de modo a torna-los inteligíveis e, por fim, carrega-os no sistema de *Data Warehouse*, para permitir um fácil acesso e análise ao conjunto dos mesmos [4].

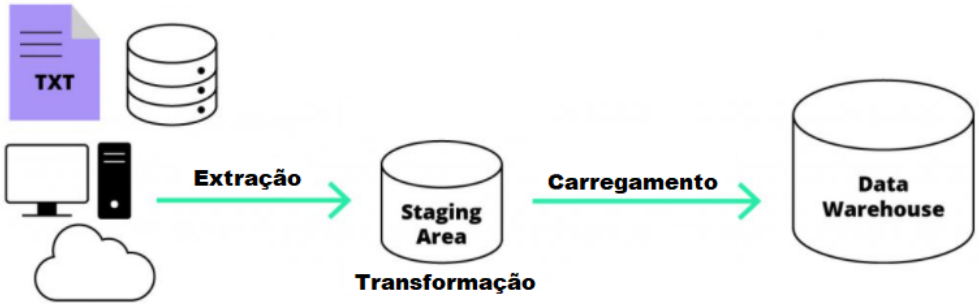

Figura 2: Processo de ETL (Adaptado de [5])

O processo de ETL divide-se em 3 etapas [4],[6]:

- Extração: Este processo de recolha, inclui a preparação necessária para a realização da integração dos dados. É de salientar ainda, que estes podem provir de diversas fontes, apresentarem diferentes formatos, ter volumes de dados distintos, entre outras inconsistências. Como tal, é essencial efetuar a consolidação de toda a informação, de modo a garantir um certo nível de consistência em todos os dados, para poderem ser alimentados no sistema e convertidos na próxima etapa. Os dados são encaminhados para uma área de transição temporária *Staging Area*, onde são organizados e convertidos para um formato único, com o objetivo de homogeneizar as diferenças existentes nas informações extraídas das diferentes fontes.

- Transformação: Envolve várias tarefas como a limpeza, filtragem e normalização dos dados, de forma a transformá-los em dados precisos, completos, consistentes e inequívocos, para estes poderem ser posteriormente importados para a Base de Dados.

- Carregamento: Os dados, após serem extraídos e transformados, são então inseridos no DW. Neste passo, é necessário carregar os dados nas dimensões e nos quadros de factos.

O ETL é considerado a peça que dá "inteligência" ao processo de *Business Intelligence*, permitindo uma visão integral de todos os pontos de atenção e variáveis envolvidas no processo de decisão.

2.4 Tableau

Tableau, **figura 3**, foi criado por Christian Chabot, Chris Stolte e Pat Hanrahan para ajudar o utilizador a visualizar dados. É uma poderosa ferramenta de exploração e descoberta de dados que permite responder a perguntas urgentes em segundos. Além de ajudar o utilizador a compreender melhor os dados, modifica a forma como se utilizam para resolver problemas, torna a sua análise mais fácil, rápida e útil.

Esta ferramenta oferece ligação a diferentes fontes de dados, como Excel, ficheiros de texto, ficheiros JSON, *MySQL*, entre outros e, após a ligação dos dados, o utilizador pode criar uma visualização dos mesmos. Também permite combinar múltiplas vistas num único dashboard interativo, proporcionando uma visão mais completa dos dados e a descoberta de conhecimento oculto em tempo real [7].

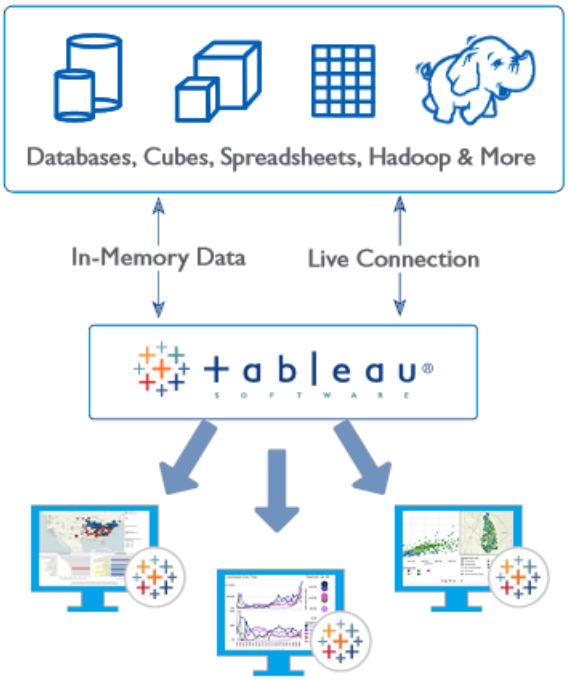

Figura 3: *Tableau* (Adaptado de [8])

3 Contextualização

A aplicabilidade da *Business Intelligence* no setor da saúde está a aumentar, uma vez que esta solução pode resolver problemas como a falta de acesso e a capacidade de utilização de dados recolhidos a partir de sistemas de informação tradicionais, e apoiar processos de tomada de decisão. Através da utilização de aplicações de BI em bases de dados de cuidados de saúde, permitiu aos profissionais de saúde concentrar os seus esforços nas áreas-chave do sistema com as maiores oportunidades de melhoria.

O DataSet em estudo contém dados fictícios relativos à doença Covid-19, a qual se encontra bem presente na atualidade e desta forma, ser importante o seu estudo.

Neste trabalho pretende-se projetar e implementar um *Data Warehouse*, com base na informação contida numa base de dados *NoSQL*, neste caso o *MongoDB*. Para tal, procedeu-se a um carregamento inicial dos dados, de um ficheiro csv, no para poderem ser, posteriormente, extraídos daí.

O primeiro passo deste estudo foi analisar cuidadosamente os dados e, para se ser capaz de compreender facilmente as relações entre colunas, foi necessário construir um modelo relacional, o que tornou possível o estudo das chaves primárias e estrangeiras do projeto. Depois da construção do DW e após todas as transformações dos dados e consecutivo carregamento destes, os dados ficam em condições de serem visualizados a partir de plataformas próprias para suporte à visualização dos mesmos, neste caso, o *Tableau*.

As seguintes secções descreverão todas as etapas necessárias para alcançar os resultados pretendidos.

4 Data Warehouse

4.1 Modelo Dimensional

Foi utilizado o *MySQL* para a criação da Base de Dados do sistema. Foi criada uma estrutura dimensional que armazena as informações recolhidas a partir dos ficheiros disponibilizados para o trabalho. De forma a implementar o *Data Warehouse* corretamente, teve-se de planear e criar o modelo Dimensional.

Um modelo dimensional usa conceitos de factos e dimensões, cujos factos são normalmente valores numéricos que podem ser agregados, e dimensões são grupos de hierarquias e descritores que definem os factos.

4.1.1 Dimensões e Factos

A definição das dimensões e factos constituintes do modelo dimensional teve por base a interpretação dos dados fornecidos. Deste modo, e sabendo que as dimensões são utilizadas com o objetivo de filtrar e categorizar os factos, foram consideradas as seguintes tabelas:

- Tabela de factos denominada "FACT_Covid19", que tem como atributos:

Atributo	Tipo de Dados	Descrição	PK	FK	NOT NULL	AI
id	INT	Identificador único da tabela	X		X	X
id_tempo	INT	Chave estrangeira para a dimensão "DIM_tempo"		X	X	
id_falta_ar	INT	Chave estrangeira para a dimensão "DIM_falta_de_ar"		X	X	
id_dor_cabeca	INT	Chave estrangeira para a dimensão "DIM_dor_cabeça"		X	X	
id_dor_muscular	INT	Chave estrangeira para a dimensão "DIM_dor_muscular"		X	X	
id_tosse	INT	Chave estrangeira para a dimensão "DIM_tosse"		X	X	
id_diarreia	INT	Chave estrangeira para a dimensão "DIM_diarreia"		X	X	
id_olfato	INT	Chave estrangeira para a dimensão "DIM_olfato"		X	X	
id_agneusia	INT	Chave estrangeira para a dimensão "DIM_agneusia"		X	X	
id_toracalgia	INT	Chave estrangeira para a dimensão "DIM_toracalgia"		X	X	
id_avalia_global	INT	Chave estrangeira para a dimensão "DIM_avalia_global"		X	X	
id_paciente	INT	Chave estrangeira para a dimensão "DIM_paciente"		X	X	
temperatura	DECIMAL (3,1)	Temperatura, em graus			X	
idade	INT	Idade, em anos			X	
duracao_sintoma	INT	Duração de cada sintoma			X	

- Dimensão "DIM_paciente", que tem como atributos:

Atributo	Tipo de Dados	Descrição	PK	FK	NOT NULL	AI
id	INT	Identificador único da tabela	X		X	X
id_paciente	INT	Número de Utente do paciente			X	
nome	VARCHAR (200)	Nome do paciente			X	
data_nasc	DATE	Data de nascimento do paciente			X	
genero	VARCHAR (10)	Género do paciente			X	
id_cod_postal	INT	Chave estrangeira para a subdimensão "SDIM_cod_postal"		X	X	

- SubDimensão "SDIM_cod_postal", que tem como atributos:

Atributo	Tipo de Dados	Descrição	PK	FK	NOT NULL	AI
id	INT	Identificador único da tabela	X		X	X
cod_postal	VARCHAR (4)	Primeiros 4 dígitos do Código Postal			X	
localidade	VARCHAR (200)	Nome da localidade			X	

- Dimensão "DIM_Tempo", que tem como atributos:

Atributo	Tipo de Dados	Descrição	PK	FK	NOT NULL	AI
id	INT	Identificador único da tabela	X		X	X
data_reg	DATE	Data de registo do sintoma			X	
fim_semana	VARCHAR (1)	Data ser ou não no fim de semana			X	
feriado	VARCHAR (1)	Data ser ou não feriado			X	
semestre	INT	Semestre correspondente			X	

- Dimensões: "DIM_avalia_global", "DIM_Falta_de_ar", "DIM_dor_cabeca", "DIM_dor_muscular", "DIM_tosse", "DIM_diarreia", "DIM_olfato", "DIM_agneusia" e "DIM_toracalgia" têm como atributos:

Atributo	Tipo de Dados	Descrição	PK	FK	NOT NULL	AI
id	INT	Identificador único da tabela	X		X	X
descricao	VARCHAR (100)	Descrição do Sintoma			X	

Depois de definidas as tabelas de dimensão e de factos, segue-se a definição do esquema dimensional, sendo que foi usado o esquema 'Estrela', isto é, um esquema em que todas as tabelas se relacionam diretamente com a tabela de factos. Neste caso, apenas uma tabela ("SDIM_cod_postal") não liga diretamente à tabela de factos, pois é uma subdimensão da dimensão "DIM_paciente".

4.1.2 Esquema Lógico

Após uma análise detalhada da estruturação do modelo dimensional, foi desenvolvido o modelo lógico que se encontra na figura seguinte, **figura 4**.

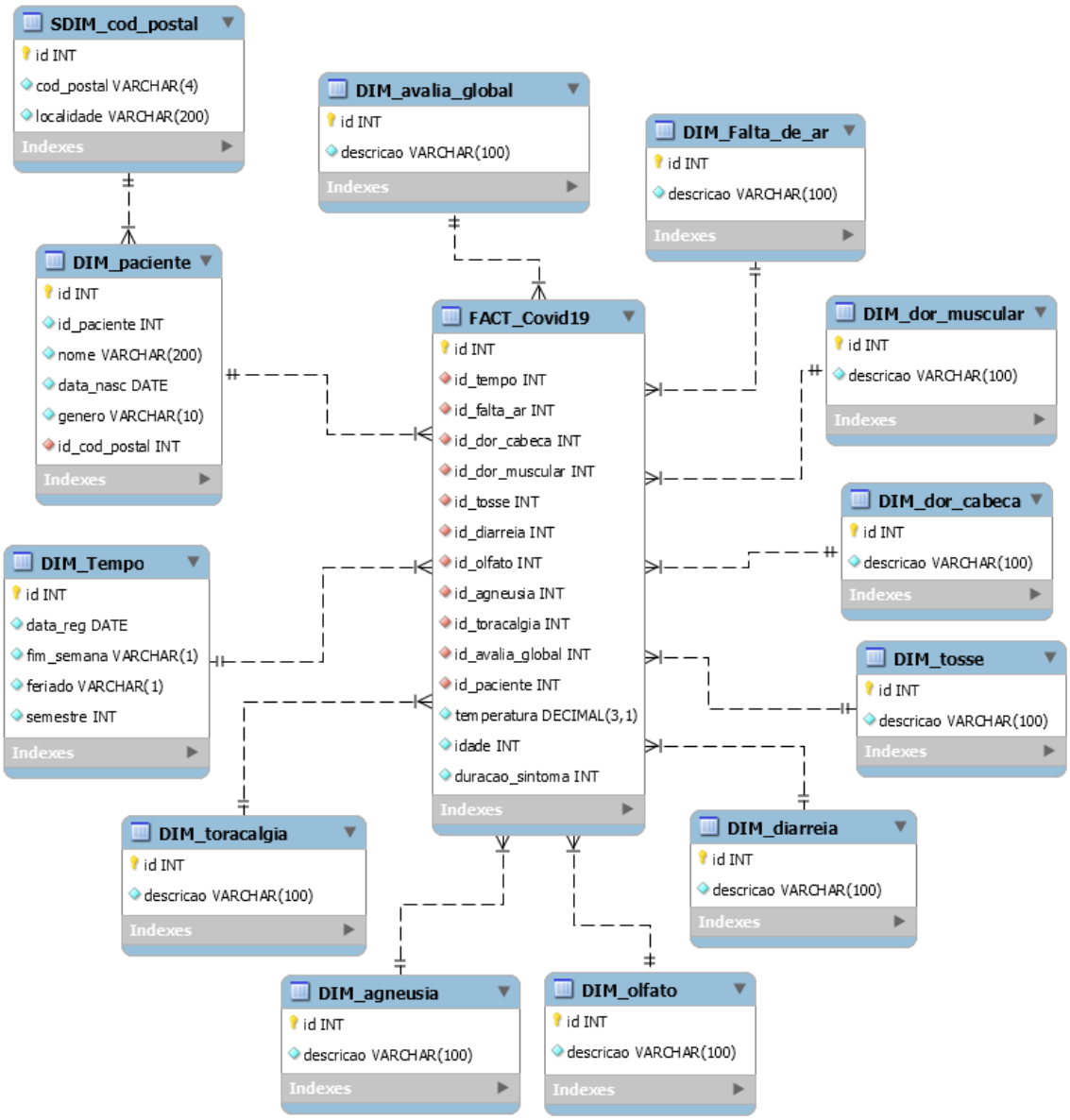

Figura 4: Modelo Lógico

Depois de criado o modelo lógico, procedeu-se à realização do *Forward Engineer*, que permitiu exportar o modelo para um servidor *MySQL*.

4.2 ETL

Uma vez definido e implementado o *Data Warehouse*, é necessário realizar o povoamento do mesmo e, para tal, foi usado o processo ETL.

Para suportar algumas fases do ETL, foi criada uma área de trabalho - "Stg_area". Trata-se de uma área de transição dos dados entre os datasets e o *Data Warehouse* final.

Assim, o processo resumiu-se a:

1. Colocar os dados de ficheiros csv no *MongoDB*, uma base de dados *NoSQL*;
2. Extrair os dados do *MongoDB* para ficheiros csv;
3. Importar os dados dos csv's para a "Stg_area";
4. Realizar as alterações/transformações necessárias aos dados;
5. Conciliar e extrair os dados da "Stg_area"para o povoamento do *Data Warehouse*.

4.2.1 MongoDB: Extração de dados

Numa primeira fase, criou-se uma base de dados "Covid19" no "*MongoDB*". Depois de criada, fez-se o carregamento dos pacientes na base de dados, de um ficheiro csv, através da utilização da função da **figura 5**. Inseridos os pacientes, procedeu-se ao carregamento, de outro csv, dos sintomas associados a cada paciente, como se pode ver na **figura 6**.

```
db.covid19.insertMany([
{ id_paciente: '1876990', nome: 'Nome do 1876990', data_nascimento: new Date('1986-02-01'), genero: 'Feminino', cod_postal: '4500', auto_avaliacao: [] },
{ id_paciente: '1877080', nome: 'Nome do 1877080', data_nascimento: new Date('1974-02-01'), genero: 'Feminino', cod_postal: '4520', auto_avaliacao: [] },
{ id_paciente: '1371762', nome: 'Nome do 1371762', data_nascimento: new Date('1966-02-01'), genero: 'Feminino', cod_postal: '4465', auto_avaliacao: [] },
{ id_paciente: '708171', nome: 'Nome do 708171', data_nascimento: new Date('1959-02-01'), genero: 'Feminino', cod_postal: '4150', auto_avaliacao: [] }
```

Figura 5: Povoamento dos pacientes na base de dados "Covid19"

```
db.covid19.update( {id_paciente: '405875'}, { $push: { auto_avaliacao: {data_reg: new Date('2020-04-01 14:07'), temperatura: '36 ºC', falta_ar: 'Não', dor_cabeca: 'Melhorou', dor_muscular: 'Não', tosse: 'Agora tenho', doarreia: '', olfato_paladar: '', agneusia: '', torocalgia: '', outros_sintomas: '', avaliacao_global: 'Estou igual'}}});
db.covid19.update( {id_paciente: '1712837'}, { $push: { auto_avaliacao: {data_reg: new Date('2020-04-01 22:13'), temperatura: '36. 5 ºC', falta_ar: 'Não', dor_cabeca: 'Não', dor_muscular: 'Não', tosse: 'Não', doarreia: '', olfato_paladar: '', agneusia: '', torocalgia: '', outros_sintomas: '', avaliacao_global: 'Sinto-me melhor'}}});
db.covid19.update( {id_paciente: '864447'}, { $push: { auto_avaliacao: {data_reg: new Date('2020-04-01 22:27'), temperatura: '37.1 ºC', falta_ar: '', dor_cabeca: '', dor_muscular: '', tosse: '', doarreia: '', olfato_paladar: '', agneusia: '', torocalgia: '', outros_sintomas: '', avaliacao_global: ''}}});
db.covid19.update( {id_paciente: '354'}, { $push: { auto_avaliacao: {data_reg: new Date('2020-04-01 23:00'), temperatura: '36.4 ºC', falta_ar: 'Não', dor_cabeca: 'Não', dor_muscular: 'Não', tosse: 'Não', doarreia: '', olfato_paladar: '', agneusia: '', torocalgia: '', outros_sintomas: '', avaliacao_global: 'Sinto-me melhor'}}});
db.covid19.update( {id_paciente: '864447'}, { $push: { auto_avaliacao: {data_reg: new Date('2020-04-01 23:01'), temperatura: '37 ºC', falta_ar: '', dor_cabeca: '', dor_muscular: '', tosse: '', doarreia: '', olfato_paladar: '', agneusia: '', torocalgia: '', outros_sintomas: '', avaliacao_global: ''}}});
```

Figura 6: Povoamento dos sintomas de cada paciente na base de dados "Covid19"

Na **figura 7** encontra-se um excerto da base de dados, já carregada com os dados.

O passo seguinte foi retirar os dados do *MongoDB*, para poderem ser usados no *MySQL Workbench*. Para isto, utilizou-se a função "Export Colection" do *MongoDB* e obteve-se um ficheiro csv com todos os pacientes e os respetivos dados pessoais. No entanto, para serem exportados os dados relativos aos sintomas por paciente, por estes estarem num Array, tiveram que ser exportados para outro ficheiro csv, através de um filtro que se correu na linha de comandos do *MongoDB*, filtro este representado na **figura 8**.

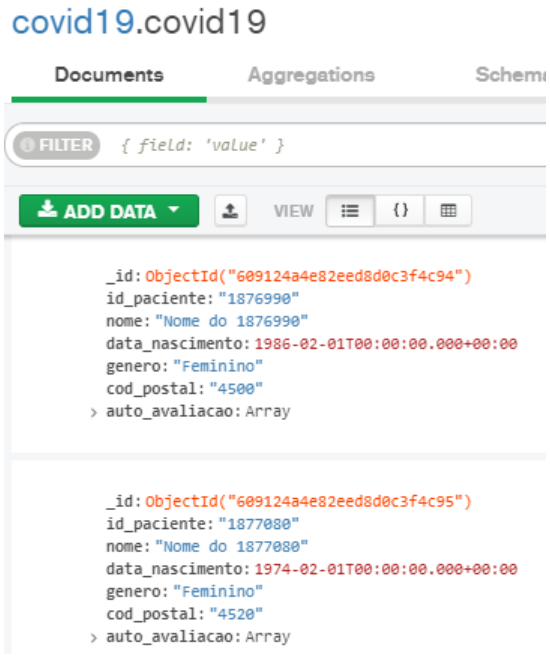

Figura 7: *MongoDB*: Base de dados "Covid19"

```
db.covid19.aggregate([{$unwind: '$auto_avaliacao'},{$project:{id_paciente:1, auto_avaliacao.data_reg":1, auto_avaliacao.falta_ar":1,
auto_avaliacao.temperatura":1, "auto_avaliacao.dor_cabeca":1, "auto_avaliacao.dor_muscular":1, "auto_avaliacao.doarreia":1,
auto_avaliacao.olfato_paladar":1, "auto_avaliacao.torocalgia":1, "auto_avaliacao.avaliacao_global":1, "auto_avaliacao.agneusia":1,
"auto_avaliacao.tosse":1, _id: 0}}, {$out:'sintomas3'}])
```

Figura 8: Filtro usado para exportar sintomas do *MongoDB*

Com este filtro foi criada uma coleção denominada de "sintomas3", que contém o paciente e os respetivos sintomas. De notar que o atributo "_id" não foi exportado, pois, sendo um identificador único correspondente à base de dados em questão, este atributo não é útil para nada fora da mesma. Depois de criada a coleção, procedeu-se da mesma forma que anteriormente para se exportar os dados para um ficheiro csv.

4.2.2 Transformação de dados

Nesta fase do processo, já com os dados extraídos do *MongoDB* e com outro dataset (códigos postais e respetivas localidades portuguesas), é necessário 'limpar' estes dados para poderem, posteriormente, serem introduzidos no *Data Warehouse*.

Antes do carregamento de todos os ficheiros csv na "Stg_area", procedeu-se a uma transformação relativa às temperaturas no ficheiro dos sintomas que foi extraído do *MongoDB*. Inicialmente as temperaturas continham inconsistências, como, por exemplo, espaços entre números ou até mesmo com as unidades escritas de forma errada. Para corrigir isto, foi elaborada uma pequena função em *python* - **figura 9** - e deste modo o atributo "temperatura"ficou coerente ao longo do dataset.

```python
import pandas as pd

def read_csv(mongo_sintomas):

    sintomas = pd.read_csv(mongo_sintomas)
    c = 0
    for value in sintomas['auto_avaliacao.temperatura']:
        c += 1
        for i in str(value):
            if i not in ['0','1','2','3','4','5','6','7','8','9','.','º','C','c',' ']:
                value_novo = str(value).replace(str(value),'')
            else:
                value_novo = str(value).replace(" ","")

        sintomas['auto_avaliacao.temperatura'] = sintomas['auto_avaliacao.temperatura'].replace([value], value_novo)

    sintomas.to_csv('mongo_sintomas.csv')

read_csv("sintomas3.csv")
```

Figura 9: Código Python para transformação do atributo "temperatura"

De seguida, fez-se a importação dos diversos datasets para a "Stg_area" através do "Table Data Import Wizard" do *MySQL Workbench*. As transformações aqui realizadas foram:

- Tratamento de palavras, de "NÃ£o" para "Não" e de "MantÃ©m" para "Mantenho", pois o csv extraído do *MongoDB* não manteve a acentuação. Para tal foi corrida a script da figura abaixo, **figura 10**.

```sql
SET SQL_SAFE_UPDATES = 0;
UPDATE mongo_sintomas
SET `auto_avaliacao.doarreia` = "Não"
WHERE `auto_avaliacao.doarreia` = 'NÃ£o';

UPDATE mongo_sintomas
SET `auto_avaliacao.torocalgia` = "Mantenho"
WHERE `auto_avaliacao.torocalgia` = 'MantÃ©m';
```

Figura 10: Excerto da script SQL para correção de problemas de acentuação

- Tratamento de datas, onde foi usada a função "left(*atributo*,10)", pois a data apenas se encontrava nos 10 primeiros caracteres e, além disto, foi necessário inverter o formato da mesma (de "%d-%m-%Y" para "%Y-%m-%d"), com o uso do "str_to_date".

- Tratamento dos valores de temperatura, pois como só se pretendia o valor sem as unidades, e este tinha um comprimento invariável foi necessário usar a função "substring", onde através da função "LOCATE", se localizou a unidade (neste caso, "ÂºC") e apenas se aproveitou o que estava para trás da mesma.

- Análise das datas de registo e calcular se estas correspondem a dias não úteis (fim de semana) ou feriado, ou a que semestre pertencem. Para isto foi usado um dataset que continha os dias de 2020 e 2021 e entre estes, que dias são dias úteis/feriados. Para processar estes dados, foi usada a função "weekday"e a função "quarter", respetivamente. Esta última função fornece o trimestre a que pertence uma determinada data (1º trimestre retorna 0, etc.), e para se saber o semestre correspondente foi usado um "if"com a condição de que, sendo o resultado desta função maior que 2, então pertence ao 2º semestre e coloca um 2, caso contrário pertence ao primeiro e coloca um 1. A função "weekday"retorna o número do dia da semana (p.e. sendo uma segunda retorna 0, uma quinta retorna 3, etc.), e da mesma forma, para fazer a verificação foi necessário usar um "if"(se o que a função retornar for maior que 4, então é fim de semana e coloca "S", caso contrário é dia útil e coloca "N").

- Tratamento dos valores nulos presentes nos diferentes parâmetros, de modo a alterar a sua representação no *Data Warehouse*. Para tal, usou-se a função "ifnull", e procedeu-se à aplicação dos seguintes critérios:

  - Valores nulos em atributos de data são substituídos por 999999;
  - Valores nulos em ids de paciente são substituídos por 99999;
  - Valores nulos em todos os restantes atributos são substituídos por 1;
  - Registos que contém uma temperatura nula não são considerados para carregamento no *Data Warehouse*, e, desta forma, são descartados.

- Para cálculo da idade e da duração do sintoma foi usada a função "TIMESTAMPDIFF" do SQL. Para o cálculo da idade (em anos) faz-se a subtração da data do registo em que se encontra com a data de nascimento. Da mesma forma, para o cálculo da duração do sintoma, em horas, é feita a diferença entre a data do primeiro registo ("min(*data*)") e a data do registo em que se encontra.

- Devido à repetição do paciente com número de utente 528665, um deles teve que ser retirado. Para isso usou-se a seguinte script, **figura 11**:

```
DELETE FROM `dw_covid`.`dim_paciente` WHERE (`id` = '88');
```

Figura 11: Script SQL para apagar linha repetida

Após todo o procedimento de transformação, verificou-se que seria importante criar 2 novas tabelas na "Stg_area", de modo que as descrições de todos os sintomas passassem a valores numéricos. Desta forma, foi criada uma tabela para todos os atributos exceto a "avalia_global". Para este atributo foi criada outra tabela, pois os resultados existentes eram diferentes. Nestas 2 tabelas, cada resultado é associado a um valor numérico.

Para a criação destas duas tabelas, foi usada a script da **figura 12**.

```
create table local_sistomas_base (id int primary key, descricao varchar(20));
create table local_sistomas_global (id int primary key, descricao varchar(20));

insert into local_sistomas_base values (0,'Desconhecido');
insert into local_sistomas_base values (1,'Não');
insert into local_sistomas_base values (2,'Melhorou');
insert into local_sistomas_base values (3,'Piorou');
insert into local_sistomas_base values (4,'Agora tenho');
insert into local_sistomas_base values (5,'Mantenho');


insert into local_sistomas_global values (0,'Desconhecido');
insert into local_sistomas_global values (1,'Estou igual');
insert into local_sistomas_global values (2,'Sinto-me melhor');
insert into local_sistomas_global values (3,'Sinto-me pior');
insert into local_sistomas_global values (4,'Não');
```

Figura 12: Script SQL para criação de 2 tabelas auxiliares

4.2.3 Povoamento do Data Warehouse

Nesta última fase, realizou-se o carregamento sequencial dos dados. É importante referir que o povoamento da tabela de factos só pode acontecer após as restantes tabelas estarem povoadas, pois este contém a chave estrangeira de cada dimensão. O código SQL usado para o efeito encontra-se em anexo.

De notar que algumas das transformações mencionadas e explicadas anteriormente, como o cálculo dos valores das idades/durações de sintomas, a substituição dos valores nulos, a verificação do semestre/dia da semana a que uma certa data de registo pertence, a seleção apenas do valor numérico da temperatura ou a exclusão dos dados com esta nula, são efetuadas juntamente com o processo de povoamento.

Esta fase destaca-se pela etapa a que habitualmente apelidam de "expectativa", uma vez que há muita expectativa para saber se todos os dados foram corretamente tratados à 'priori', se tudo funciona e é povoado conforme o suposto, ou se é necessário voltar à fase do tratamento de dados e proceder a mais alterações.

4.3 Atualização Incremental e/ou Diferencial

Uma das decisões mais importantes na implementação de um *data warehouse* é determinar como é efetuado o carregamento dos dados para garantir a consistência das informações, pois uma má decisão nesta fase pode comprometer a validade dos dados.

Geralmente, o carregamento dos dados nas tabelas de dimensões e factos podem ser de dois tipos: total ou incremental.

De forma total, como o próprio nome diz, trata-se do carregamento completo dos dados sempre que há a execução de um novo processo ETL. Desta forma, capturar as modificações nos dados de origem é crucial. Este processo pode ser feito recorrendo a *triggers*. O termo *trigger* define uma estrutura de base de dados que funciona, como o nome sugere, como uma função que é disparada mediante alguma ação. Geralmente essas ações que disparam os triggers são alterações nas tabelas por meio de operações de inserção, exclusão e atualização de dados (*insert, delete* e *update*) [9].

Pelo contrário, carregar os dados de forma incremental consideram-se apenas os novos registos dos sistemas operativos no ETL, inserindo-os ao repositório do DW.

Além da forma como é povoado, num contexto real, um *data warehouse* deve ser atualizado regularmente, de modo a garantir que as informações derivadas deste são atuais. A cada atualização ocorre o processo ETL, processo já descrito anteriormente. Isto exige maior volume de trabalho e maior consumo de tempo.

Este processo pode ser feito através de um sistema de atualização diferencial/incremental. De acordo com este, são apenas carregados os dados que não estão presentes no *data warehouse*, ou seja, corresponde à inserção de novas informações ou à substituição das mesmas.

Começando pelo cenário em que se pretende introduzir nova informação. Para o efeito, em vez de utilizar o conjunto de dados original utiliza-se um novo conjunto de dados que possui apenas a informação a ser inserida, sendo, para isto, necessário estabelecer uma conexão à base de dados. Estabelecida a conexão, é realizado, de novo, todo o processo de ETL para os novos dados. É importante ter em atenção que este conjunto novo de dados deve possuir os mesmos campos do original. Encontrando-se a informação consistente, deve ser efetuado um carregamento sequencial dos dados começando, como sempre, dos elementos mais simples (dimensões) para os mais complexos (factos).

Passando agora para uma perspetiva de atualização dos dados, deve ser estabelecida uma conexão à base de dados que suporta o *data warehouse* de forma iterar sobre as dimensões e factos. À semelhança do que foi dito anteriormente, em vez de se utilizar conjunto de dados

original, utiliza-se um novo conjunto de dados que possui apenas a informação a atualizar. Mais uma vez passa-se pelo processo ETL sendo especialmente importante o tratamento dos valores nulos de cada linha. Posteriormente, deve verificar-se, para cada linha desse conjunto, se a mesma está presente na dimensão a que se destina. Se estiver, faz-se a substituição da antiga pela atualizada. Caso contrário, esta é adicionada à respetiva dimensão. Por fim, deve ser atualizada a tabela de factos sendo também necessário verificar se o valor que está atualmente na tabela de factos corresponde ao da linha a analisar. Caso não corresponda, é efetuada a atualização da tabela de factos com o respetivo valor [10].

Posto isto, pode-se concluir que o carregamento total tem a vantagem de poder ser tratado como uma longa transação em lote que, quando termina, produz um novo repositório, ao mesmo tempo que possibilita o processamento de consultas dos dados do repositório corrente, durante a transação. Contudo, o carregamento total pode exigir um longo período de transação, mesmo quando se utilizam as técnicas de paralelismo. Por outro lado, o carregamento incremental lida com transações de carregamento mais pequenas para evitar entrar em conflito com o processamento das consultas aos dados do *data warehouse*, exigindo, assim, que a sequência das transações seja coordenada, de modo a assegurar a consistência dos dados derivados e dos índices relativamente aos dados base [11].

5 Business Intelligence

Com recurso ao *Tableau*, foi possível criar indicadores que irão fornecer a este estudo uma forma de analisar se variáveis como o género, a idade, a localidade, a duração do sintoma e as suas diferentes descrições estão relacionadas entre si, ou com os diversos pacientes.

Foram realizados oito gráficos, considerados relevantes para o caso em estudo, que serão apresentados nas secções abaixo.

5.1 Variação de temperatura em relação à média por Género e por Idade

Para implementar o gráfico da **figura 13** foi calculado o desvio padrão, em cada tipo de género, para cada valor de idade. Além disto, é apresentada a linha média, de forma, a poder visualizar em qual das idades o valor de desvio padrão, em relação à média, foi superior.

Pela análise do gráfico, consegue-se perceber que, apesar do sexo feminino apresentar maiores valores de desvio relativamente à média, é o sexo masculino que apresenta um range de maiores desvios ao longo de diferentes idades. Por esse motivo, pode-se concluir que os homens têm tendência a apresentar maiores variações de temperatura que as mulheres e, dessa forma, consegue-se tomar diferentes precauções consoante o paciente seja homem ou mulher.

Quanto à idade, tanto no sexo masculino como feminino, as maiores variações de temperatura ocorrem para a idade de 20 e a partir dos 70/75 anos, evidenciando o maior valor de variação para a idade 100 no sexo feminino. Desta forma, mostra-se que é necessário, para pacientes com estas idades, tomar mais medidas preventivas.

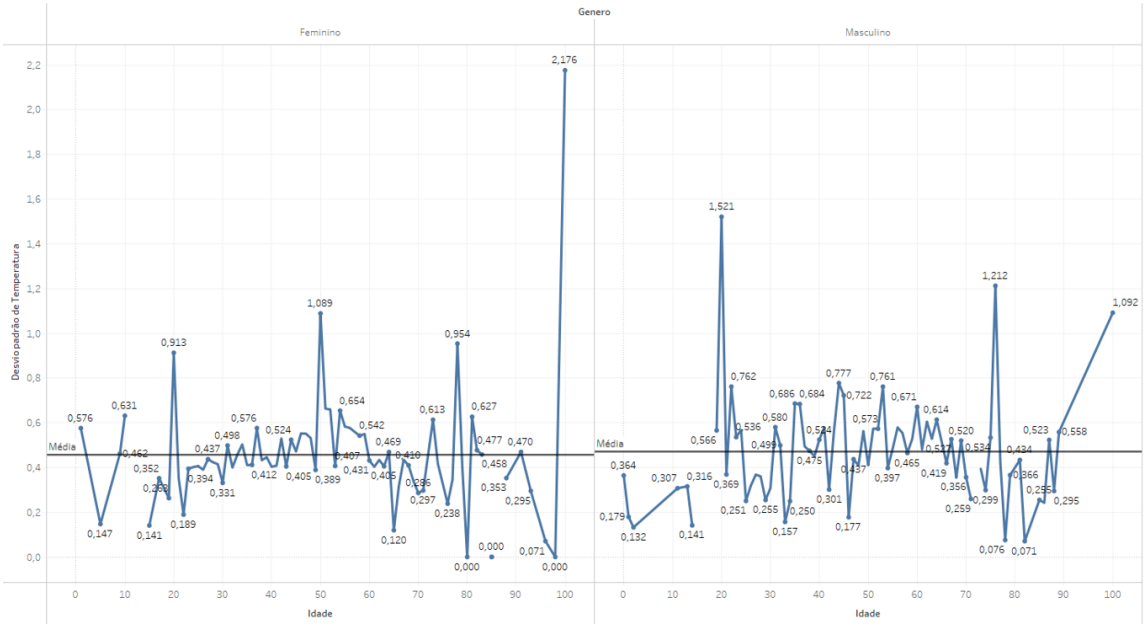

Figura 13: Variação de temperatura em relação à média por Género e por Idade

5.2 Nº de pacientes por Localidade

Quais são as localidades que apresentam maior número de pacientes com sintomas?

Com o gráfico da **figura 14**, é possível chegar à conclusão que diferentes cidades apresentam uma certa quantidade de pacientes com sintomas. Através desta informação, consegue-se tomar medidas como confinamento ou medidas de restrição de mobilidade, tendo em conta a quantidade de pacientes com sintomas que é superior a um determinado valor de *threshold*. As cidades a salientar são: Porto, Vila Nova de Gaia, Jovim e Rio Tinto.

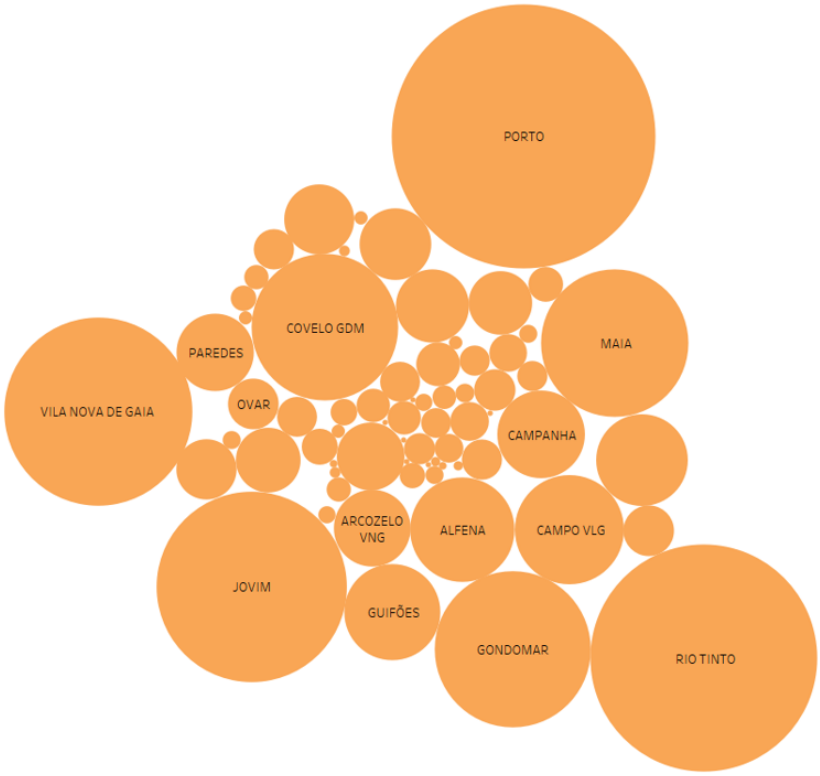

Figura 14: Nº de pacientes por Localidade

5.3 Nº de pacientes por Idade

Que idades são mais afetadas pelo Covid-19?

O gráfico da **figura 15** tenta responder à pergunta acima referida, constatando que o maior número de pacientes regista-se entre os 20 e os 70 anos de idade, sendo os 27, 37, 57 e 62 anos com maior número de pacientes com Covid-19. Com esta informação, é possível tomar em consideração, que pessoas dentro desta gama de idades serão, à partida, mais suscetíveis de possuir Covid-19. Desta forma, diferentes medidas poderão ser tomadas tendo em conta a idade do paciente.

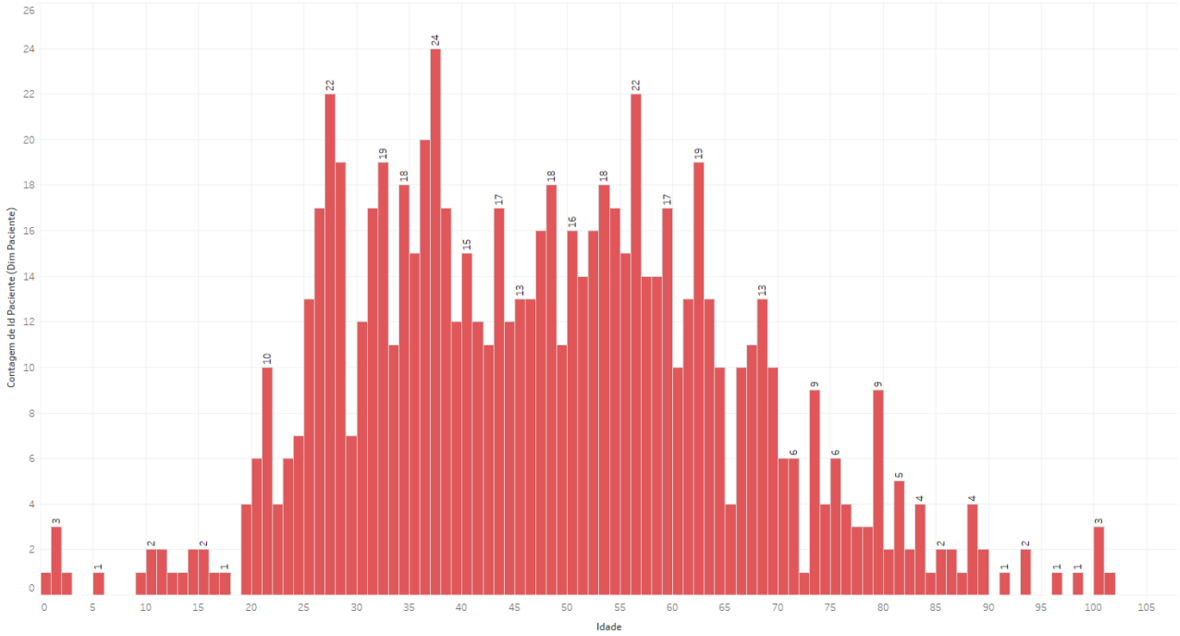

Figura 15: Nº de pacientes por Idade

5.4 Nº de pacientes que apresentam registo em fim de semana ou feriado

Há mais pacientes com registos a um fim de semana ou a um feriado?

O gráfico da **figura 16**, com uma escala logarítmica no eixo y para melhor compreensão dos dados, evidencia se as pessoas tendem a realizar testes ao Covid-19 e registar sintomas, preferencialmente, nos fins de semana ou durante a semana e se o fazem ou não durante os feriados.

Através da análise do gráfico abaixo, chega-se à conclusão que os pacientes, preferencialmente, preferem ser testados durante a semana em vez de no fim de semana. Além disso, como seria de esperar, constata-se que as pessoas realizam o teste em dias que não são feriados, já que existem mais dias que não são feriados do que aqueles que são.

Tendo em conta este conhecimento, qualquer hospital ou clínica pode tomar medidas, nomeadamente no que diz respeito à distribuição, número de 'staff' ou armazenamento de material para os testes, de forma a possuir mais recursos durante a semana e durante dias que não sejam feriado do que o contrário. Assim, consegue-se obter uma melhor eficiência e eficácia na distribuição dos testes, por exemplo.

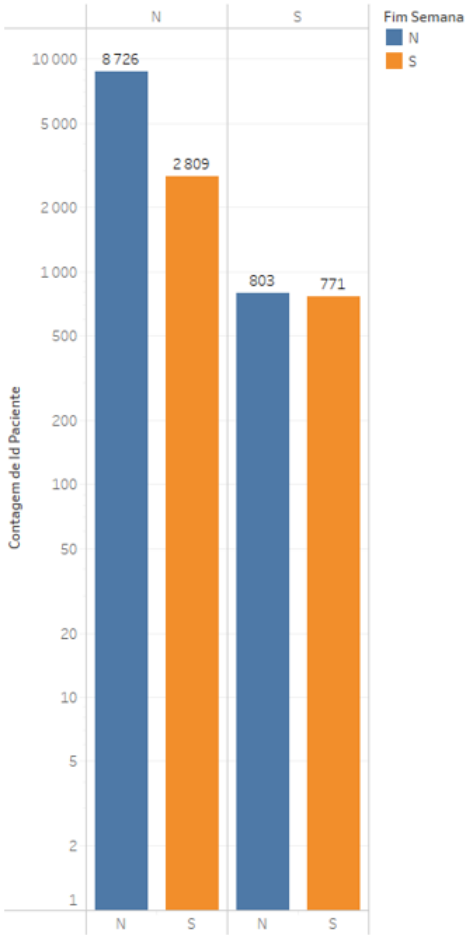

Figura 16: Nº de pacientes que apresentam registo num fim de semana e/ou feriado

5.5 Nº de pacientes em cada classificação da avaliação global por Género

Será que o género está relacionado com os diferentes valores de classificação para a métrica da avaliação global?

Através de uma análise cuidada do gráfico da **figura 17**, consegue-se concluir que, de facto, o sexo está relacionado com os diferentes valores de classificação para a métrica da avaliação global. Isto acontece, uma vez que existem mais pacientes do sexo feminino do que masculino em qualquer um dos tipos de classificação para a avaliação global. Tal significa, que algum ou alguns sintomas de Covid-19 poderão estar intrinsecamente relacionados com o sexo feminino, podendo, desta forma, tomar as devidas precauções em detrimento do sexo do paciente.

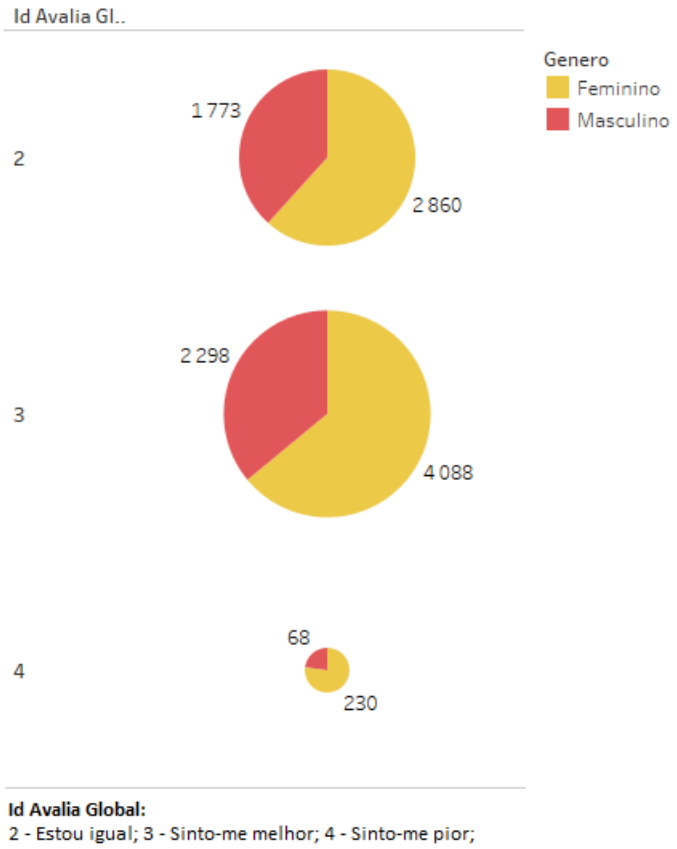

Figura 17: Nº de pacientes em cada classificação da avaliação global por Género

5.6 Média de duração do sintoma por Género e por Idade

A duração do sintoma está intimamente relacionada com o género e/ou idade do paciente?

Com o recurso ao gráfico da **figura 18**, infere-se que existe uma relação entre ambos (duração do sintoma e género/idade).

Comparando, por um lado, pacientes com sexo feminino e masculino, conclui-se que existem mais picos de duração para o sexo feminino. Desta forma, mostra-se que os homens, poderão ter alguma maior resistência a um ou a vários sintomas de Covid-19 relativamente às mulheres.

Por outro lado, analisando a idade com a duração do sintoma, comprova-se que para o sexo masculino, a gama de maior picos encontra-se entre os 45 e 75 anos de idade. No entanto, existe duas exceções para um paciente com, aproximadamente, 90 anos de idade e um paciente recém-nascido. Relativamente ao sexo feminino, a gama de maior picos encontra-se entre os 20 e os 70 anos de idade, tendo, da mesma forma que o sexo masculino, duas exceções em pacientes com 90 anos de idade e recém-nascidos.

Desta forma, consegue-se, novamente, tomar pedidas de precaução, sabendo a idade de qualquer paciente e o seu sexo, já que se pode ter uma ideia da duração do(s) sintoma(s).

Figura 18: Média de duração do sintoma por Género e por Idade

5.7 Média de duração do sintoma e Idade por Localidade

Existe alguma relação entre as pessoas com mais idade de uma determinada localidade e a sua correspondente duração de sintoma?

Através da análise do gráfico da **figura 19**, pode-se concluir que, de facto, existe uma relação. Tal acontece, uma vez que, de uma forma geral, localidades que apresentam uma média de idade superior à média, também apresentam uma média de duração de sintoma superior à média. Com este conhecimento, torna-se possível alertar localidades cujos habitantes apresentam idades superiores à média, por exemplo, do país, uma vez que, poderão com elevada probabilidade, desenvolver sintomas que acabarão por durar mais tempo que a média.

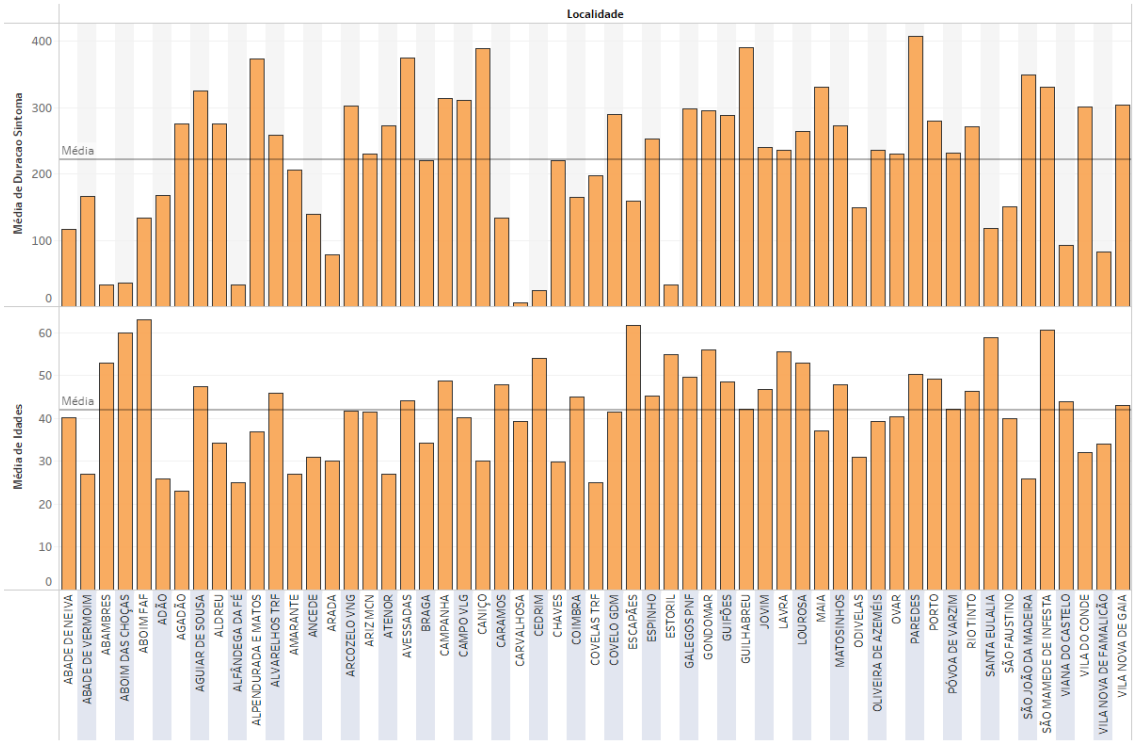

Figura 19: Média de duração do sintoma e média de Idade por Localidade

5.8 Nº de pacientes em cada valor de classificação para diversos sintomas

Será que o número de pacientes com um determinado valor de classificação para a tosse é o mesmo para outros sintomas, como a dor de cabeça, a diarreia ou a falta de olfato?

Analisando o gráfico da **figura 20**, conclui-se que as mulheres apresentam, de uma forma geral, um maior número para o valor de classificação de "melhorou", "piorou", "agora tenho" e "mantenho" em qualquer um dos sintomas relativamente aos homens, com especial atenção ao sintoma "perda de olfato", no qual a diferença é ainda mais significativa. Desta forma, verifica-se que estas estão mais correlacionadas com os sintomas de Covid-19.

Observando dentro de cada género, constata-se que no sexo feminino, existem sempre mais pacientes que não apresentam qualquer sintoma. De salientar, que a "perda de olfato" demonstra maiores valores de melhoria, mostrando que as mulheres conseguem resistir bem a este sintoma. Por outro lado, os homens também apresentam mais pacientes sem qualquer sintoma e também no sintoma "perda de olfato" apresentam mais pacientes que "melhoraram", tal como no sintoma "tosse". Assim, é possível compreender que os homens, tal como as mulheres, possuem melhores mecanismos de resistência ao sintoma "perda de olfato" e "tosse" do que a outros sintomas.

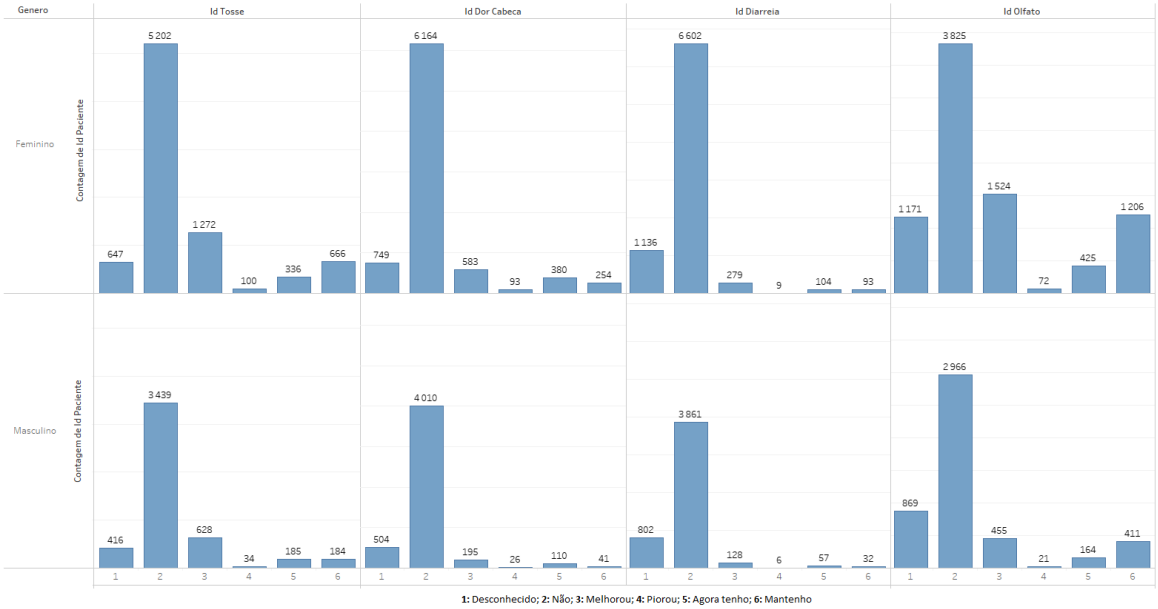

Figura 20: Nº de pacientes em cada valor de classificação para diversos sintomas

6 Conclusão

O presente relatório descreveu o processo de implementação de um sistema de *Data Warehousing*, bem como um sistema de *Business Intelligence*, para suporte à decisão clínica.

Embora estas ferramentas tragam várias vantagens para as organizações de saúde, há um trabalho árduo antes da sua construção. Todos os dados têm de passar por um sistema ETL, ou seja, a informação extraída tem de ser tratada antes de ser carregada para garantir não só a qualidade dos dados mas também a sua fiabilidade.

Por fim, o *Tableau* revelou-se uma ferramenta útil, pois com esta ferramenta foi possível criar indicadores e filtros com diferentes gráficos e estatísticas. Ajudou a analisar o conjunto de dados de uma forma visual, mais fácil e rápida do que olhar para a própria base de dados.

Este projeto incentivou, não só a aprendizagem e exploração relativa à análise de dados, como também mitigou o manuseamento de ferramentas como o *MongoDB*, *MySQL* e *Tableau*, que foram necessários para a elaboração do trabalho.

É de salientar que foram encontradas dificuldades, especialmente durante o povoamento dos dados na DW, pois, tal como dito anteriormente, é uma etapa de "expectativa"e foi preciso recuar várias vezes ao tratamento dos dados para poder concluir este processo.

Os resultados finais vão ao encontro das expectativas iniciais do projeto, uma vez que no final foi possível adquirir conhecimento relativamente à relação entre os diferentes sintomas do Covid-19 e com a idade, sexo ou localidade dos pacientes.

Em suma, considera-se que o trabalho produzido, apesar das adversidades encontradas durante a sua realização, cumpre com os todos os requisitos mencionados pelo docente no enunciado do mesmo.

Referências

[1] SNS24. "COVID-19," acedido a: 24 maio, 2021. [Online]. Available: https://www.sns24.gov.pt/tema/doencas-infecciosas/covid-19/

[2] Higor Medeiros. "Introdução ao MongoDB," acedido a: 22 maio, 2021. [Online]. Available: https://www.devmedia.com.br/introducao-ao-mongodb/30792#Caracteristicas

[3] Wikidot. "Data Warehouse e Data Mining," acedido a: 21 maio, 2021. [Online]. Available: http://tic-gmcm-ed4.wikidot.com/dwh

[4] Guru99. "ETL (Extract, Transform, and Load) Process in Data Warehouse," acedido a: 22 maio, 2021. [Online]. Available: https://www.guru99.com/etl-extract-load-process.html

[5] MJV Team. "O que é ETL e por que devemos integrar dados?," acedido a: 21 maio, 2021. [Online]. Available: https://www.mjvinnovation.com/pt-br/blog/o-que-e-etl-como-funciona/

[6] João Ferreira, Miguel Miranda, António Abelha e José Machado, "O Processo ETL em Sistemas Data Warehouse," Janeiro 2010. [Online]. Available: https://www.researchgate.net/publication/265195317_O_Processo_ETL_em_Sistemas_Data_Warehouse

[7] Guru99. "What is Tableau? Uses of Tableau Software Tool," acedido a: 22 maio, 2021. [Online]. Available: https://www.guru99.com/what-is-tableau.html

[8] WisdomAxis. "Explain Tableau Architecture or Framework?," acedido a: 24 maio, 2021. [Online]. Available: https://www.wisdomaxis.com/technology/software/tableau/interview-questions/explain-tableau-architecture.php

[9] Joel Rodrigues. "Triggers no SQL Server: teoria e prática aplicada em uma situação real," acedido a: 26 maio, 2021. [Online]. Available: https://www.devmedia.com.br/triggers-no-sql-server-teoria-e-pratica-aplicada-em-uma-situacao-real/28194

[10] Diego Elias. "Cargas no Data Warehouse - Total ou Incremental?," acedido a: 26 maio, 2021. [Online]. Available: https://canaltech.com.br/business-intelligence/data-science-a-evolucao-do-data-analytics-como-aliado-de-customer-experience-162380/

[11] Carlos Manuel Rogado Quintino da Costa Paiva, "O processo de refrescamento nos sistemas de Data Warehouse," M.S. thesis, Instituto Superior de Ciências do Trabalho e da Empresa, Lisboa, 2006.

Anexos

A. Povoamento do Data Warehouse

```
#Povoar Subdimensão "sdim_cod_postal"
insert into dw_covid.sdim_cod_postal(cod_postal,localidade)
select * from stg_area.codigo_postal;

#Povoar Dimensão "dim_paciente"
insert into dw_covid.dim_paciente(id_paciente,nome,data_nasc,genero,id_cod_postal)
select id_paciente,nome, left(data_nascimento,10),genero,
ifnull((select id from dw_covid.sdim_cod_postal a2 where a1.cod_postal=a2.cod_postal),1)  cod_postal
from stg_area.pacientes a1;

#Povoar Dimensão "dim_agenusia"
insert into dw_covid.dim_agneusia (descricao)
select descricao from stg_area.local_sistomas_base;

#Povoar Dimensão "dim_diarreia"
insert into dw_covid.dim_diarreia (descricao)
select descricao from stg_area.local_sistomas_base;

#Povoar Dimensão "dim_dor_cabeca"
insert into dw_covid.dim_dor_cabeca (descricao)
select descricao from stg_area.local_sistomas_base;

#Povoar Dimensão "dim_dor_muscular"
insert into dw_covid.dim_dor_muscular (descricao)
select descricao from stg_area.local_sistomas_base;

#Povoar Dimensão "dim_Falta_de_ar"
insert into dw_covid.dim_Falta_de_ar (descricao)
select descricao from stg_area.local_sistomas_base;



#Povoar Dimensão "dim_olfato"
insert into dw_covid.dim_olfato (descricao)
select descricao from stg_area.local_sistomas_base;

#Povoar Dimensão "dim_toracalgia"
insert into dw_covid.dim_toracalgia (descricao)
select descricao from stg_area.local_sistomas_base;

#Povoar Dimensão "dim_tosse"
insert into dw_covid.dim_tosse (descricao)
select descricao from stg_area.local_sistomas_base;

#Povoar Dimensão "dim_avalia_global"
insert into dw_covid.dim_avalia_global (descricao)
select descricao from stg_area.local_sistomas_global;

#Povoar Dimensão "dim_tempo"
insert into dw_covid.dim_tempo (data_reg,fim_semana,feriado,semestre)
select str_to_date(Data,'%d-%m-%Y'),
if(weekday(str_to_date(Data,'%d-%m-%Y'))>4,'S','N') fim_semana,Feriado,
if(QUARTER(str_to_date(Data,'%d-%m-%Y'))>2,2,1) semestre
from stg_area.calendario_dim_tempo;
```

```
#Povoar Tabela de Factos "fact_covid19" - ÚLTIMA TABELA A SER POVOADA
insert into dw_covid.fact_covid19(id_tempo, id_falta_ar, id_dor_cabeca, id_dor_muscular, id_tosse, id_diarreia,
    id_olfato, id_agneusia, id_toracalgia, id_avalia_global, id_paciente, temperatura, idade, duracao_sintoma)
select
ifnull((select id from dw_covid.dim_tempo a2 where left(a1.`auto_avaliacao.data_reg`,10)=a2.data_reg),999999) id_tempo,
ifnull((select id from dw_covid.dim_falta_de_ar a2 where a1.`auto_avaliacao.falta_ar`=a2.descricao),1) id_falta_ar,
ifnull((select id from dw_covid.dim_dor_cabeca a2 where a1.`auto_avaliacao.dor_cabeca`=a2.descricao),1) id_dor_cabeca,
ifnull((select id from dw_covid.dim_dor_muscular a2 where a1.`auto_avaliacao.dor_muscular`=a2.descricao),1) id_dor_muscular,
ifnull((select id from dw_covid.dim_tosse a2 where a1.`auto_avaliacao.tosse`=a2.descricao),1) id_tosse,
ifnull((select id from dw_covid.dim_diarreia a2 where a1.`auto_avaliacao.doarreia`=a2.descricao),1) id_diarreia,
ifnull((select id from dw_covid.dim_olfato a2 where a1.`auto_avaliacao.olfato_paladar`=a2.descricao),1) id_olfato,
ifnull((select id from dw_covid.dim_agneusia a2 where a1.`auto_avaliacao.agneusia`=a2.descricao),1) id_agneusia,
ifnull((select id from dw_covid.dim_toracalgia a2 where a1.`auto_avaliacao.torocalgia`=a2.descricao),1) id_toracalgia,
ifnull((select id from dw_covid.dim_avalia_global a2 where a1.`auto_avaliacao.avaliacao_global`=a2.descricao),1) id_avalia_global,
ifnull((select id from dw_covid.dim_paciente a2 where a1.id_paciente=a2.id_paciente),99999) id_paciente,
substring(`auto_avaliacao.temperatura`,1,LOCATE('Â ºC',`auto_avaliacao.temperatura`)-1) temperatura,
TIMESTAMPDIFF(YEAR, (select data_nasc from dw_covid.dim_paciente a2 where a1.id_paciente=a2.id_paciente),
    left(a1.`auto_avaliacao.data_reg`,10)) idade,
TIMESTAMPDIFF(HOUR,(select min(left(a2.`auto_avaliacao.data_reg`,10)) from stg_area.mongo_sintomas a2
    where a1.id_paciente=a2.id_paciente),left(a1.`auto_avaliacao.data_reg`,10)) duracao_sintoma
from stg_area.mongo_sintomas a1 where `auto_avaliacao.temperatura`!='';
```