

Universidade do Minho

Mestrado Integrado em Engenharia Informática
Aplicações Informáticas na Biomedicina

**A realização de urgências gerais num determinado hospital
nacional**

GRUPO 13
Nuno Silva
A78156

4 de Janeiro de 2020

Conteúdo

1	Introdução	1
2	Data Warehouse	2
3	Povoamento	3
3.1	Cursors	3
3.2	Talend	5
3.3	Cursors vs Talend	8
4	Power BI	8
5	Inclusão dos indicadores clínicos	11
6	Conclusão	12

1 Introdução

Nesta unidade curricular, foi proposto aos alunos a realização de um trabalho prático com o tema ”**A realização de urgências gerais num determinado hospital nacional**”, de forma a consolidar os conhecimentos e ferramentas expostas no decorrer das aulas. Assim sendo foi disponibilizado, pela equipa docente, um ficheiro num formato *.csv*, contendo dados reais de urgências gerais num determinado hospital nacional.

Neste ficheiro encontramos dados, tais como:

- URG_EPISODIO - Identificador único da urgência;
- DATAHORA_ADM - Registo temporal da hora de admissão do utente;
- DATAHORA_ALTA - Registo temporal da hora da alta atribuída ao utente;
- ALTA_DES_ESPECIALIDADE - Descrição da especialidade;
- DES_LOCAL - Descrição do local;
- DES_PROVENIENCIA - Descrição da proveniência;
- SEXO - Género do paciente;
- DTA_NASCIMENTO - Data de nascimento do paciente;
- DES_CAUSA - Descrição do motivo pela admissão do utente nas urgências;

Num primeira fase, este dados serão utilizados para povoar um *data warehouse*. Este mesmo será efetuado via cursores e com jobs em *Talend*.

Por fim, para uma análise mais visual, foi utilizada a ferramenta *Power BI* de forma a criar alguns indicadores clínicos, que posteriormente poderão vir a ser utilizados numa aplicação informática.

2 Data Warehouse

Após uma análise sobre *dataset* entregue, foi possível a criação do seguinte *Data Warehouse*:

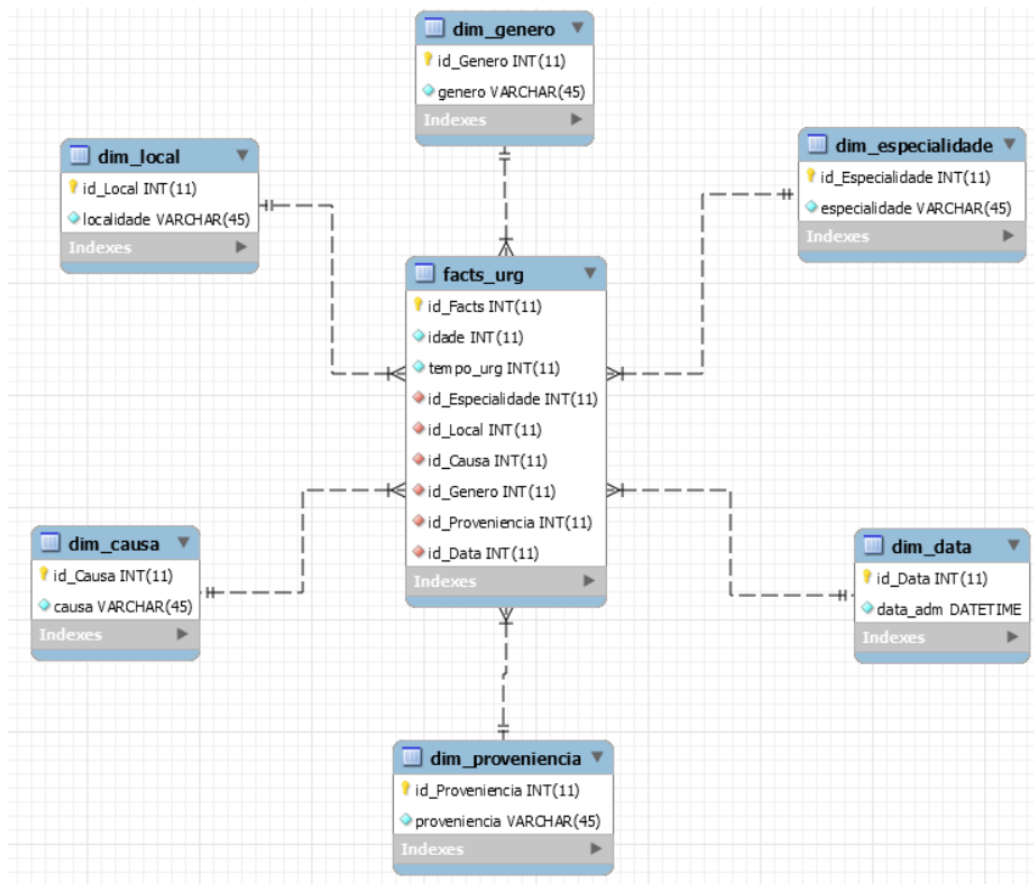


Figura 1: Data Warehouse em esquema de estrela

Numa primeira análise sobre o modelo, verificamos que campos como data nascimento ou data de alta, não estão presentes em nenhuma das tabelas em apresentadas. Estas não são apresentadas por exemplo na tabela, *DIM_DATA* devido a apenas a data de admissão importar para este caso concreto. Contudo estes registos não são desprezados, e são utilizados para efetuar o cálculo da idade e do tempo em que um utente se encontrou nas urgências (*tempo_urg*).

De notar que todos os *IDs* presentes nas tabelas são auto incrementáveis, de formar a não surgir problemas no momento em que será povoado este *data warehouse*.

3 Povoamento

De forma a fazer o povoamento deste *data warehouse*, tal como referido anteriormente, foram utilizados dois métodos, cursores em SQL e jobs no programa Talend.

3.1 Cursores

Para povoar o *data warehouse* em questão, foi escolhida a criação de cursores utilizando a ferramenta *MySQL Workbench*. Os dados inseridos, foram retirados de um *schema* auxiliar, denominado bd_urg, para onde foram importados os dados do ficheiros csv.

Assim foram criados cursores para cada tabela do *data warehouse*. De seguida, é mostrado o *script* responsável pela povoação da tabela DIM_Local.

```
DELIMITER //
CREATE DEFINER='root'@'localhost' PROCEDURE `insereLocal`()
BEGIN
    -- Variaveis
    DECLARE feito BOOLEAN DEFAULT FALSE;
    DECLARE localV VARCHAR(75);
    -- Cursor
    DECLARE curs CURSOR FOR SELECT DISTINCT DES_LOCAL FROM bd_urg.urg_inform_geral;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET feito = TRUE;

    OPEN curs;

    -- Loop
    get_loop: LOOP
        FETCH curs INTO localV;

        IF feito THEN
            LEAVE get_loop;
        END IF;

        -- Preenche a tabela em questao
        INSERT INTO DW_URG.dim_local (localidade) VALUES (localV);

    END LOOP;

    CLOSE curs;
END //
```

Tal como será de esperar, todas as outras tabelas têm um raciocínio semelhante, pelo que não se justifica fazer aqui demonstração do código. Contudo, para a tabela de factos, temos um cursor com algumas diferenças, uma vez que é necessário inserir dados com chaves estrangeiras definidas anteriormente. Para além de que, é necessário processar dados de forma a obter a idade de um utente e o tempo que ele passou na urgência do hospital.

```
DELIMITER //
CREATE DEFINER='root'@'localhost' PROCEDURE 'insereFactsUrg'()
BEGIN
  -- Variaveis
  DECLARE feito BOOLEAN DEFAULT FALSE;
  DECLARE id INT;
  DECLARE id_CAUSA INT;
  DECLARE id_PROVENIENCIA INT;
  DECLARE id_DATA INT;
  DECLARE id_ESPECIALIDADE INT;
  DECLARE id_GENERO INT;
  DECLARE id_LOCAL INT;
  DECLARE idade INT;
  DECLARE tempoUrg INT;

  -- Cursors
  DECLARE curs CURSOR FOR
    SELECT UGERAL.URG_EPISODIO, TIMESTAMPDIFF(YEAR,UGERAL.
      DTA_NASCIMENTO, UGERAL.DATAHORA_ALTA), TIMESTAMPDIFF(
      MINUTE, UGERAL.DATAHORA_ADM, UGERAL.DATAHORA_ALTA), DC.
      id_Causa, DD.id_Data, DE.id_Especialidade, DG.id_Genero, DL.id_Local, DP.
      id_Proveniencia
    FROM dw_urg.dim_causa AS DC, dw_urg.dim_data AS DD, dw_urg.dim_especialidade
      AS DE, dw_urg.dim_genero AS DG, dw_urg.dim_local AS DL, DW_URG.
      dim_proveniencia AS DP, bd_urg.urg_inform_geral AS UGERAL
    WHERE DC.causa = UGERAL.DES_CAUSA AND DD.data_adm = UGERAL.
      DATAHORA_ADM AND DE.especialidade = UGERAL.
      ALTA_DES_ESPECIALIDADE AND DG.genero = UGERAL.SEXO AND DL.
      localidade = UGERAL.DES_LOCAL AND DP.proveniencia = UGERAL.
      DES_PROVENIENCIA ;

  DECLARE CONTINUE HANDLER FOR NOT FOUND SET feito = TRUE;

  OPEN curs;

  get_loop: LOOP
    FETCH curs INTO id, idade, tempoUrg, id_CAUSA, id_DATA, id_ESPECIALIDADE,
      id_GENERO, id_LOCAL, id_PROVENIENCIA;

    IF feito THEN
      LEAVE get_loop;
```

END IF;

```
INSERT INTO dw_urg.facts_urg(id_Facts, idade, tempo_urg, id_Especialidade, id_Local
, id_Causa, id_Genero, id_Proveniencia, id_Data)
VALUES(id, idade, tempoUrg, id_ESPECIALIDADE, id_LOCAL, id_CAUSA,
id_GENERO, id_PROVENIENCIA, id_DATA);
```

END LOOP;

CLOSE curs;

END //

3.2 Talend

Fazendo seguimento da linha de pensamento utilizada anteriormente, primeiro de tudo foram povoadas as tabelas de dimensão, e no fim a tabela de factos.

Assim sendo foi criado o seguinte *job*, na ferramenta **Talend**:

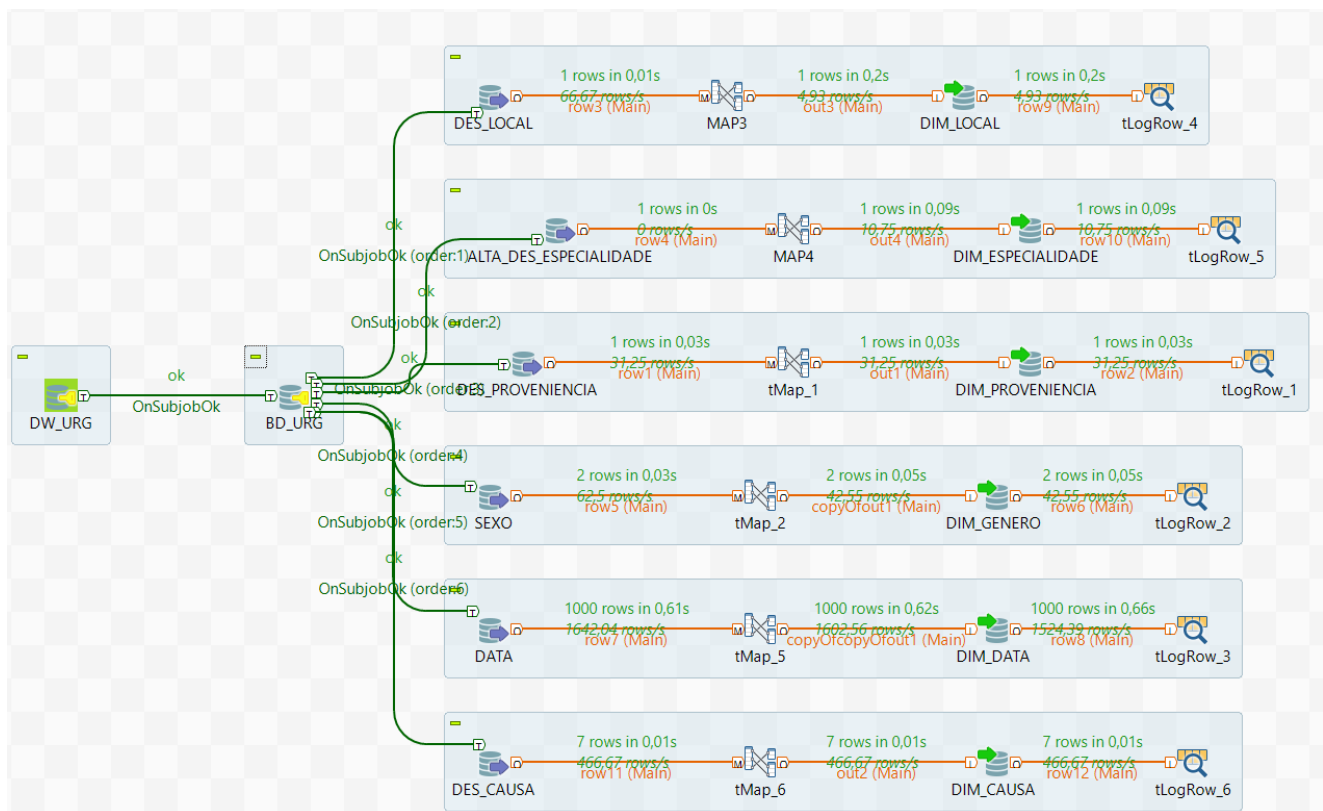


Figura 2: Povoamento de tabelas usando *jobs*

De seguida é demonstrado a *querie* utilizada para povoar a tabela "SEXO", do *schema* BD_URG. Claro sendo que as restantes tabelas, são povoadas de forma semelhante.

```
SELECT 'urg_inform_geral'.SEXO FROM 'urg_inform_geral'
```

Posteriormente, o resultado desta *querie* é enviado para um **tMap**, que por sua vez realizará o mapeamento necessário, para que no fim o resultado seja enviado para a tabela pretendida.

De forma a assegurar *IDs* e por sua vez diferentes entre si, foi utilizada a seguinte expressão, durante a utilização de um **tMap**.

```
Numeric.sequence("id_Causa",1,1)
```

Onde podemos verificar o seu efeito, por exemplo, num *output* de um **tLogRow**.

tLogRow_6	
causa	id_Causa
DOENCA	1
ACIDENTE DE TRABALHO	2
OUTRAS	3
ACIDENTE ESCOLAR	4
ACIDENTE PESSOAL	5
ACIDENTE DE VIACAO	6
QUEDA	7

Figura 3: Output de um **tLogRow**

Por fim e com todas as tabelas povoadas, foi criado um job com a função de povoar a tabela *Facts_Urg*, este povoamento, tal como nos cursores, será diferente pois necessitamos dos identificadores únicos, *IDs*, das restantes tabelas. Para além do cálculo da idade de um paciente, e o tempo, em minutos, que passou nas urgências do hospital.

De forma a obter um cálculo correto tanto da idade como o tempo nas urgências, foram utilizadas as seguintes expressões:

- Cálculo da idade:

```
Math.round(TalendDate.diffDate(TalendDate.getCurrentDate(),
row11.DTA_NASCIMENTO, "dd") / 365 )
```

- Cálculo do tempo, em minutos:

```
Math.round(((row11.DATAHORA_ALTA.getTime() - row11.DATAHORA_ADM.
getTime()) / (60 * 1000)) % 60 )
```

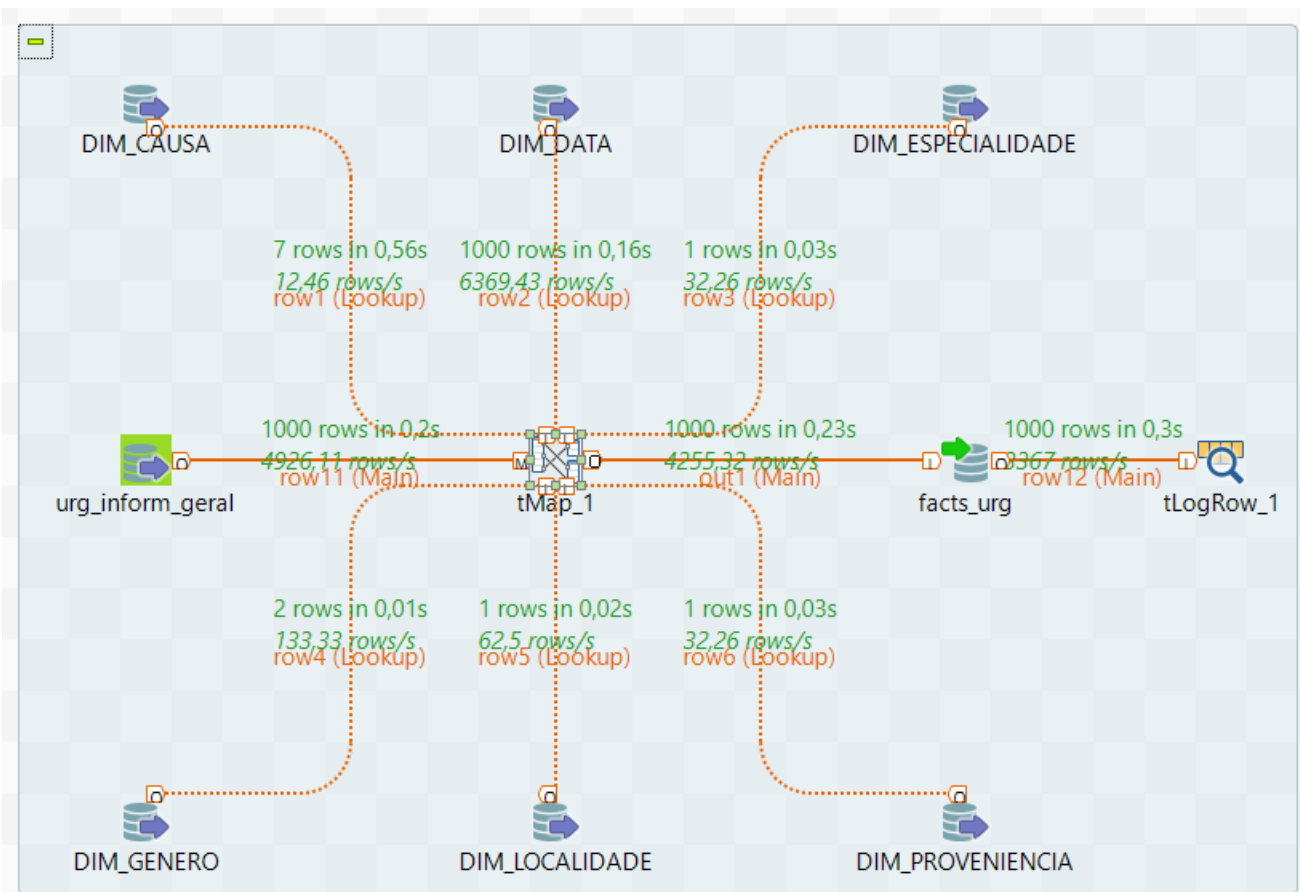



Figura 4: Job que irá povoar a tabela Facts_Urg

Sendo que este povoamento também poderemos verificar na seguinte imagem:

tLogRow_1								
id_Facts	idade	tempo_urg	id_Especialidade	id_Local	id_Causa	id_Genero	id_Proveniencia	id_Data
15051576	27	60	1	1	1	1	1	1
15051577	32	140	1	1	1	1	1	2
15051578	84	219	1	1	1	1	1	3
15051579	61	64	1	1	1	1	1	4
15051580	63	59	1	1	1	2	1	5
15051581	30	78	1	1	1	1	1	6
15051582	88	116	1	1	1	2	1	7
15051583	41	56	1	1	1	1	1	8
15051584	67	49	1	1	1	2	1	9
15051585	4	53	1	1	1	2	1	10
15051586	59	111	1	1	1	1	1	11
15051587	10	61	1	1	1	2	1	12
15051588	83	92	1	1	1	1	1	13
15051589	86	66	1	1	1	2	1	14
15051590	20	108	1	1	1	2	1	15
15051591	73	107	1	1	1	1	1	16
15051592	78	180	1	1	1	1	1	17
15051593	77	108	1	1	1	2	1	18
15051594	50	108	1	1	1	2	1	19

Figura 5: Excerto do povoamento da Facts_Urg

3.3 Cursores vs Talend

Para o povoamento, via cursores SQL, foi evidente uma monotonia deste mesmo, pois todo um processo para de povoamento de uma tabela, por exemplo DIM_Local, será equivalente as restantes, exceptuando a tabela Facts_URG. Este processo requeriu muito mais tempo relativamente aos outros, sendo que teve de haver uma atenção extra para os detalhes.

Por ventura, usando a ferramenta Talend, todo o processo tornou-se muito mais agil e simplificado. Pois tornou o processo de povoamento muito mais rápido. Este método não há necessidade em haver uma preocupação com código.

Assim sendo, verifica-se que a utilização do Talend em detrimento ao SQL, é uma boa opção.

4 Power BI

O *Power BI* é uma ferramenta que nos fornece visualizações interativas com um interface simples e de fácil compreensão para o utilizador. Permitindo também a ligação com diversas fontes de informação e, posteriormente, gerar novo conhecimento por via de relatórios e *dashboards*.

Assim sendo foram criados alguns indicadores clínicos, primeiro de tudo escolhemos aqueles relacionados com o género. Tais como número de pessoas do sexo masculino e feminino, com base nas causas que os levaram a frequentar as urgências e no tempo que em média cada género frequenta as urgências.

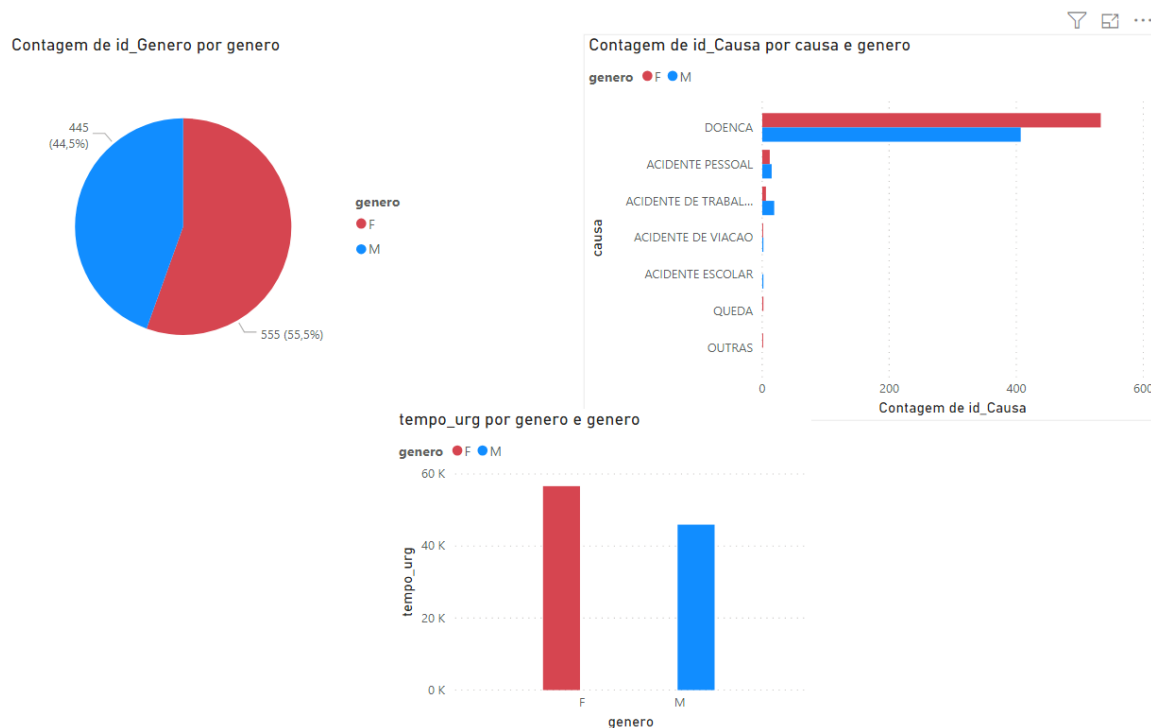


Figura 6: Indicadores clínicos relacionados com género

Posteriormente foram gerados indicadores, mais relacionados nas médias e medianas, como por exemplo a mediana do tempo que cada causa leva nas urgências, ou contagem das ocorrências de cada causa. Este mesmos indicadores, apresentam uma análise mais detalhada sobre o que se esta a passar numa urgência, e por ventura uma melhor compreensão sobre problemas que poderão levar a um maior tempo espera.

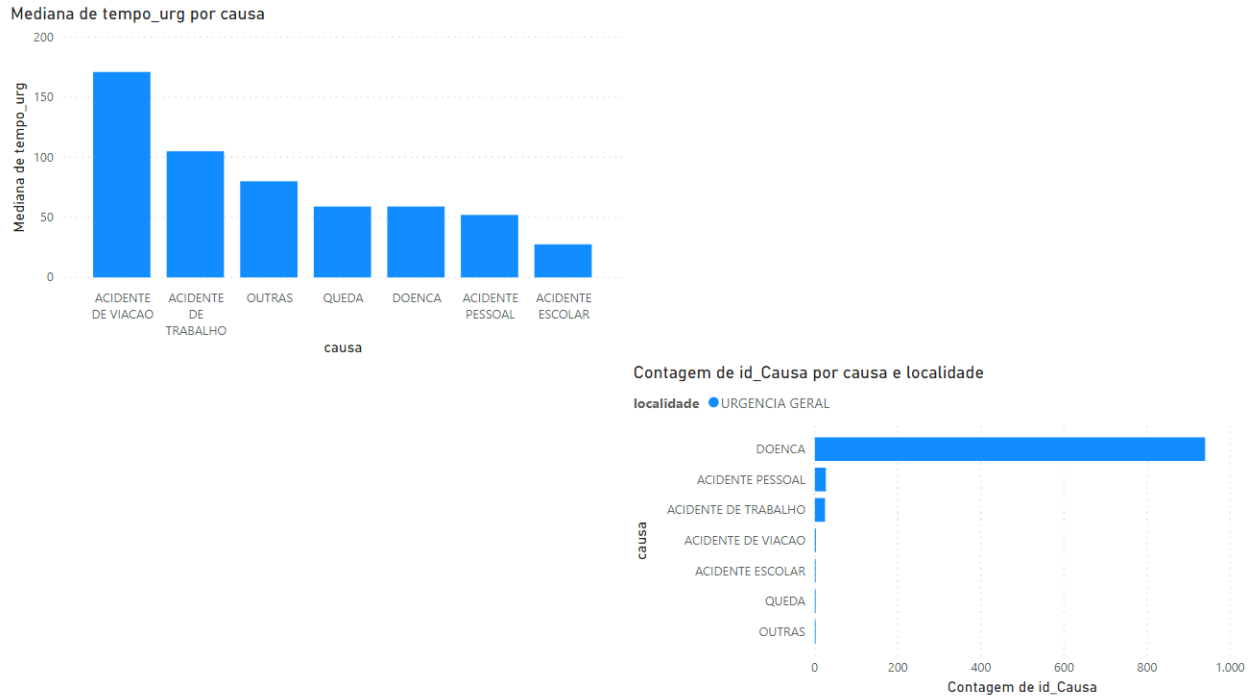


Figura 7: Indicadores clínicos relacionados com causas

Por fim, e tendo em consciência que neste conjunto de dados apenas se encontram registos do mês de Janeiro, foram criados indicadores relacionados com o dia, causa e o tempo total na urgência, para além da média de idades por causa.

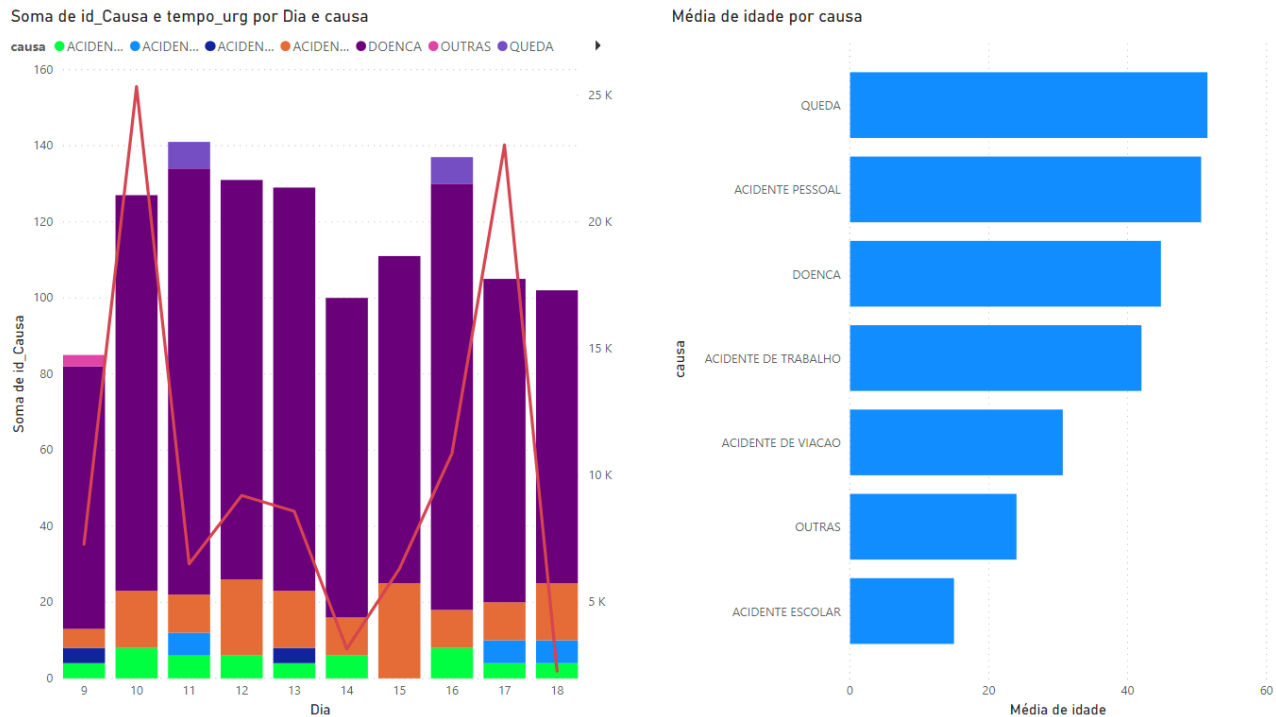
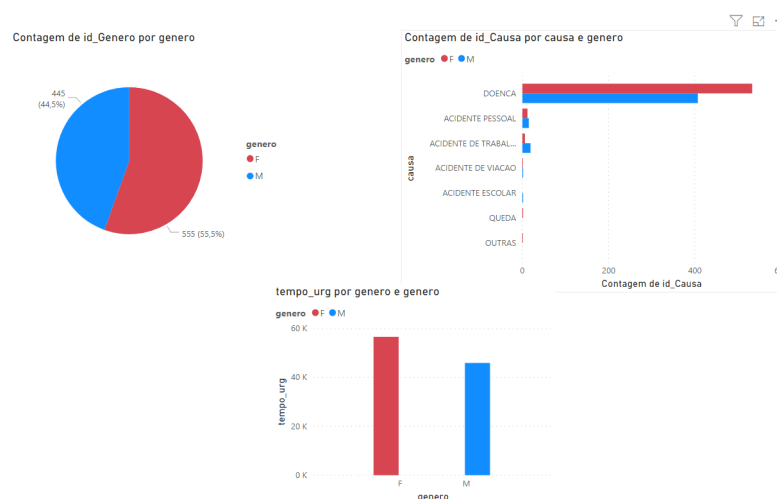


Figura 8: Indicadores clínicos relacionados com tempo e idade

De seguida é apresentada a *dashboard*:



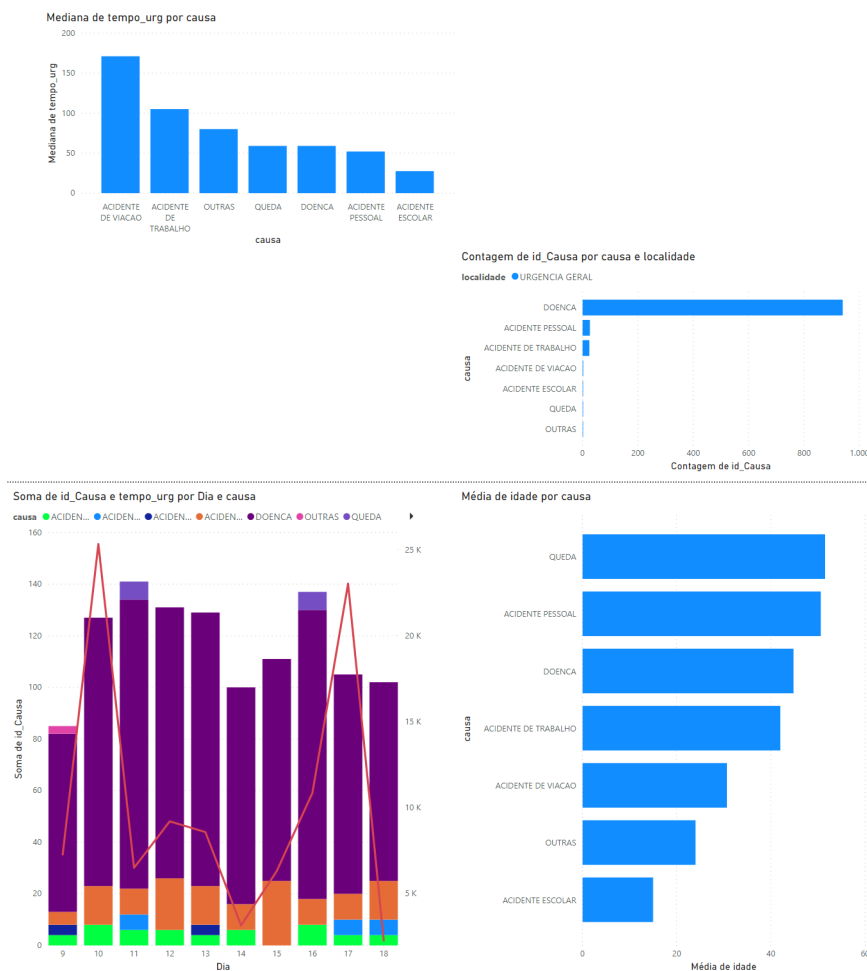


Figura 9: Dashboard

5 Inclusão dos indicadores clínicos

De forma a tornar estes indicadores ainda mais interactivos, poderemos utiliza-los numa aplicação informática.

Para tal, a aplicação poderá demonstrar os dados por tópicos, onde cada página poderia ter a informação representada sob a forma de gráficos ou tabelas.

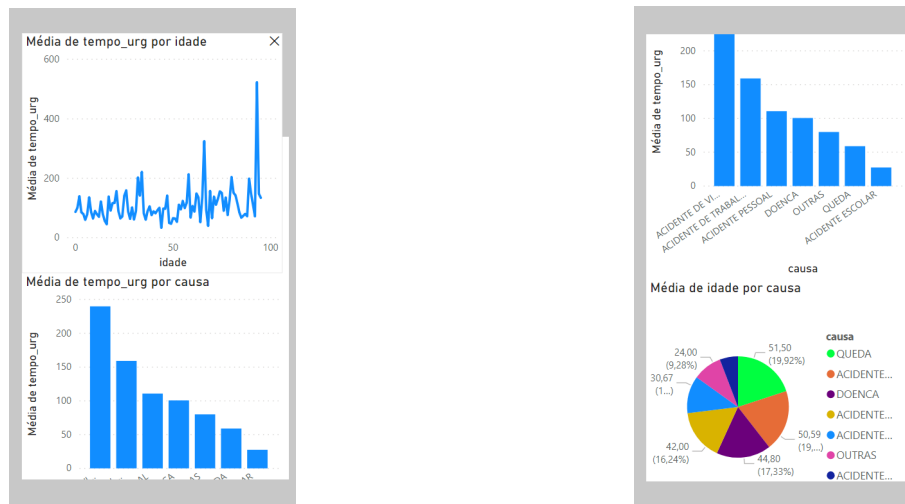


Figura 10: Exemplo da utilização de indicadores em aplicações informáticas

6 Conclusão

Com a realização deste trabalho prático, concluo que consegui adquirir todos os conhecimentos abordados desta unidade curricular. A construção de um *data warehouse* e os diferentes métodos para a sua povoação.

Por fim, fico satisfeito com o trabalho realizado, e com a percepção de terem sido atingidas todas as metas pretendidas. Como trabalho futuro, ficaria a criação de mais indicadores clínicos, otimização no processo de povoamento do *data warehouse*, e o desenvolvimento da aplicação que iria utilizar os indicadores criados.