

UNIVERSIDADE DO MINHO

APLICAÇÕES INFORMÁTICAS NA BIOMEDICINA

---

## Trabalho Prático - Urgências Hospitalares

---

*Autores:*

Daniel Vieira  
João Palmeira  
José Ramos

*Números Aluno:*

A73974  
A73864  
A73855

17 de Dezembro de 2019

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Desenvolvimento</b>	<b>4</b>
2.1	<i>MySQL</i>	4
2.1.1	Modelo Dimensional	4
2.1.2	Exemplo Tabela de Dimensão	6
2.1.3	Tabela de Factos	7
2.2	<i>Talend</i>	11
2.2.1	Povoamento	12
2.2.2	Conexão às Bases de Dados	12
2.2.3	Tabelas de Dimensão	13
2.2.4	Tabelas de Factos	15
2.3	<i>MySQL</i> vs <i>Talend</i>	18
2.4	Indicadores Clínicos <i>POWER BI</i>	19
2.4.1	Média de tempo de alta do Paciente por Causa	20
2.4.2	Média de tempo de alta do Paciente por Género	21
2.4.3	Contagem por Género	21
2.4.4	Idade do Paciente por Causa	22
2.4.5	Média de idade do Paciente por Causa	23
2.4.6	Ocorrências de Causas	24
2.4.7	Género por Causa	25
2.5	<i>Dashboard</i>	26
2.5.1	<i>Dashboard:</i> "Causas"	26
2.5.2	<i>Dashboard:</i> "Géneros"	27
2.6	<i>Mockups</i>	28
2.6.1	Aplicações Web	28
2.6.2	Aplicações Móveis	29
<b>3</b>	<b>Conclusões</b>	<b>30</b>

## Lista de Figuras

1	Dados do <i>Schema</i> BD_URG . . . . .	4
2	Modelo dimensional DW_URG em formato estrela . . . . .	5
3	Extrato da geração do Modelo Físico . . . . .	6
4	<i>Procedure</i> "PREENCHE_DIM_DATA()" . . . . .	7
5	Variável finais do povoamento da tabela de Factos . . . . .	7
6	Variável tabela de dimensão para cursos . . . . .	8
7	Declaração dos cursos da Data e da Especialidade . . . . .	8
8	Ciclo externo . . . . .	9
9	Ciclo interno . . . . .	9
10	Atribuição das variáveis finais . . . . .	10
11	<i>Job</i> . . . . .	11
12	Ligaçāo a Base de Dados dentro da Job . . . . .	12
13	Recolha da informação para o Talend da base de dados bd_urg. . . . .	13
14	Inserção dos dados obtidos na Tabela de Dimensão respetiva. . . . .	14
15	Recolha da informação para o Talend da base de dados bd_urg. . . . .	16
16	<i>DB_Output_Facts</i> . . . . .	17
17	<i>Output Schema</i> . . . . .	17
18	Base de dados no POWER BI . . . . .	19
19	Média de tempo de alta do Paciente por Causa . . . . .	20
20	Média de tempo de alta do Paciente por Género . . . . .	21
21	Contagem por Género . . . . .	21
22	Idade do Paciente por Causa . . . . .	22
23	Média de idade do Paciente por Causa . . . . .	23
24	Ocorrências de Causas . . . . .	24
25	Género por Causa . . . . .	25
26	<i>Dashboard</i> : "Causas" . . . . .	26
27	<i>Dashboard</i> : "Géneros" . . . . .	27
28	Género por Causa . . . . .	28
29	Género por Causa . . . . .	29

## 1 Introdução

Na Unidade Curricular **Aplicações Informáticas na Biomedicina** foi-nos proposto o desenvolvimento de um trabalho com as ferramentas abordadas tanto nas aulas práticas como as teóricas, sendo estas o **MySQL**, **Talend** e o **POWER BI**.

Para este trabalho, foi-nos fornecido um ficheiro de dados CSV chamado **inform\_geral.csv** de forma a podermos manuseá-lo de modo a responder aos requisitos pedidos no enunciado.

A informação está já disponibilizada com vários campos de modo a que possamos analisá-los de acordo com a nossa finalidade de conseguir obter informação de um **Data Warehouse** no contexto da informática na biomedicina.

Além do resultado final que se pretende, já especificado antes, este deve ser acompanhado com uma relatório sucinto, técnico e claro de modo a poder explicar todos os passos efetuados, explicando-os bem como a avaliação crítica dos resultados obtidos.

## 2 Desenvolvimento

### 2.1 MySQL

Primeiramente foi-nos disponibilizado um ficheiro no formato ”.csv”, ”**urg\_inform\_geral.csv**”, respetivo ao *dataset* que contém os dados reais da lista de realização de urgências gerais num determinado hospital nacional.

Tal como nos foi proferido, a informação representada no *dataset* contém informação referente a 1000 ocorrências e disponibiliza dados, tais como, o número de episódio da urgência (identificador único), a data e a hora da admissão do paciente nas urgências, a data e a hora da alta, a descrição da especialidade da alta, a descrição do local, a descrição da proveniência, a descrição da causa da entrada nas urgências, o género do paciente e a sua data de nascimento.

De modo a inicializar o tratamento de dados e tal como foi pedido no enunciado, foi necessário transferir a informação do ficheiro ”**urg\_inform\_geral.csv**” para o MySQL. Como tal criámos um novo *schema* no *MySQL Workbench* denominado ”**BD\_URG**” e importamos todos os dados do *dataset* através da opção ”**Table Data Import Wizard**”.

```

1 • use bd_urg;
2 • select * from urg_inform_geral;

```

URG_EPISODIO	DATAHORA_ADM	DATAHORA_ALTA	ALTA_DES_ESPECIALIDADE	DES_LOCAL	DES_PROVENIENCIA	SEXO	DTA_NASCIMENTO	DES_CAUSA
15051576	2017-01-09 13:24:06	2017-01-09 14:24:29	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1992-12-26	DOENCA
15051577	2017-01-09 13:25:44	2017-01-09 15:46:43	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1987-04-02	DOENCA
15051578	2017-01-09 13:33:30	2017-01-09 17:13:27	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1935-05-26	DOENCA
15051579	2017-01-09 13:35:43	2017-01-09 14:39:43	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1958-11-08	DOENCA
15051580	2017-01-09 13:50:28	2017-01-09 14:50:18	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	1956-09-08	DOENCA
15051581	2017-01-09 13:56:01	2017-01-09 15:14:11	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1989-03-05	DOENCA
15051582	2017-01-09 14:00:49	2017-01-09 15:57:32	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	1931-07-28	DOENCA
15051583	2017-01-09 14:03:52	2017-01-09 15:00:02	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1979-01-07	DOENCA
15051584	2017-01-09 14:17:38	2017-01-09 15:07:02	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	1952-09-16	DOENCA
15051585	2017-01-09 14:21:45	2017-01-09 15:15:11	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	2015-03-15	DOENCA
15051586	2017-01-09 14:23:01	2017-01-09 16:14:26	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	1960-03-16	DOENCA
15051587	2017-01-09 14:24:22	2017-01-09 15:25:05	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	2009-12-21	DOENCA
15051588	2017-01-09 14:28:42	2017-01-09 16:01:06	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1936-03-30	DOENCA
15051589	2017-01-09 14:31:01	2017-01-09 15:38:01	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	1933-04-28	DOENCA
15051590	2017-01-09 14:37:34	2017-01-09 16:26:15	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	1999-10-08	DOENCA
15051591	2017-01-09 14:44:58	2017-01-09 16:32:05	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1947-01-06	DOENCA
15051592	2017-01-09 14:47:20	2017-01-09 17:47:20	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1941-09-17	DOENCA
15051593	2017-01-09 14:48:41	2017-01-09 16:36:52	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	1942-08-16	DOENCA
15051594	2017-01-09 14:52:01	2017-01-09 16:40:26	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	M	1969-06-13	DOENCA
15051595	2017-01-09 15:01:24	2017-01-09 17:22:47	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1948-11-24	DOENCA
15051596	2017-01-09 15:04:04	2017-01-09 16:05:08	CLINICA GERAL	URGENCIA GERAL	EXTERIOR	F	1953-05-12	DOENCA

Figura 1: Dados do *Schema BD\_URG*

#### 2.1.1 Modelo Dimensional

Depois da importação dos dados para o *schema* criado, é necessário desenhar e criar um modelo dimensional no formato de esquema em estrela no *MySQL Workbench - EER Diagram*, numa base de dados denominada de ”**DW\_URG**”. Para corresponder as exigências decidiu-se, para a criação deste modelo, definir seis **Tabelas de Dimensão** e uma **Tabela de Factos**:

- **DIM.LOCAL**
- **DIM.GENERO**

- DIM\_PROVENIENCIA
- DIM\_ESPECIALIDADE
- DIM\_DATA
- DIM\_CAUSA
- FACTS\_URG

Através destas máximas foi possível desenhar o diagrama tal como é apresentado na figura seguinte:

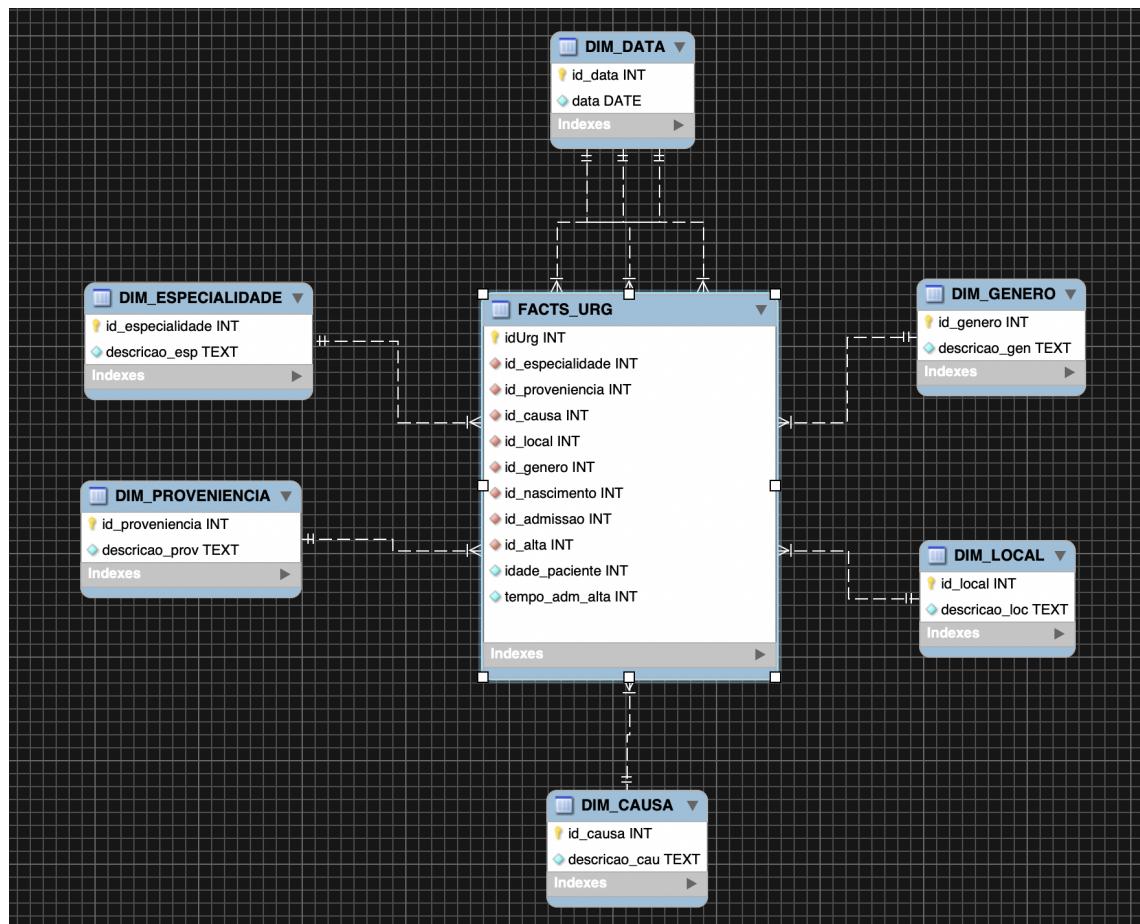


Figura 2: Modelo dimensional DW\_URG em formato estrela

De seguida, tal como foi pedido no enunciado, foi gerado o **Modelo Físico**, associado ao modelo dimensional definido anteriormente, através da opção "**Forward Engineer**".

```

13  -- Schema DW_URG
14  -----
15  CREATE SCHEMA IF NOT EXISTS `DW_URG` DEFAULT CHARACTER SET utf8 ;
16  USE `DW_URG` ;
17
18  -----
19  -- Table `DW_URG`.`DIM_ESPECIALIDADE`
20  -----
21  DROP TABLE IF EXISTS `DW_URG`.`DIM_ESPECIALIDADE` ;
22
23  ⊖ CREATE TABLE IF NOT EXISTS `DW_URG`.`DIM_ESPECIALIDADE` (
24      `id_especialidade` INT NOT NULL AUTO_INCREMENT,
25      `descricao_esp` TEXT NOT NULL,
26      PRIMARY KEY (`id_especialidade`)
27  ENGINE = InnoDB;
28
29
30  -----
31  -- Table `DW_URG`.`DIM_PROVENIENCIA`
32  -----
33  DROP TABLE IF EXISTS `DW_URG`.`DIM_PROVENIENCIA` ;
34
35  ⊖ CREATE TABLE IF NOT EXISTS `DW_URG`.`DIM_PROVENIENCIA` (
36      `id_proveniencia` INT NOT NULL AUTO_INCREMENT,
37      `descricao_prov` TEXT NOT NULL,
38      PRIMARY KEY (`id_proveniencia`)
39  ENGINE = InnoDB;
40
41
42  -----
43  -- Table `DW_URG`.`DIM_CAUSA`
44  -----
45  DROP TABLE IF EXISTS `DW_URG`.`DIM_CAUSA` ;
46
47  ⊖ CREATE TABLE IF NOT EXISTS `DW_URG`.`DIM_CAUSA` (
48      `id_causa` INT NOT NULL AUTO_INCREMENT,
49      `descricao_cau` TEXT NOT NULL,
50      PRIMARY KEY (`id_causa`))

```

Figura 3: Extrato da geração do Modelo Físico

Com o Modelo Físico gerado, é necessário povoar este novo *data warehouse*. Para o povoamento vai ser utilizado os dados da tabela ”urg\_inform\_geral” da base de dados ”BD\_URG”, preenchendo as tabelas de Dimensão e, por último, a tabela de factos através de **Procedures** e **Cursosores**.

### 2.1.2 Exemplo Tabela de Dimensão

Segundo o que foi mencionado anteriormente, o preenchimento das tabelas de Dimensão e da tabela de factos é realizado através de *procedures* e cursosres. Deste modo, visto que o processo é semelhante para todas as tabelas, passamos a apresentar um exemplo deste povoamento.

Numa primeira fase, para a criação de um *procedure*, são declaradas as variáveis que provém da base de dados ”BD\_URG” e, de seguida, cria-se o cursor que irá percorrer as linhas de uma coluna e guardar os dados na respetiva tabela de dimensão.

Tal como podemos ver no exemplo seguinte, o cursor irá percorrer, através do ciclo ”**get\_data: LOOP**”, todos as linhas com dados da coluna ”**v\_desDatas**” e guardar esses dados na tabela de Dimensão ”**DIM\_DATA**” até que o ciclo atinja o seu fim em ”**END LOOP get\_data**”.

```

1 •   CREATE DEFINER='root'@'localhost' PROCEDURE `PREENCHE_DIM_DATA`()
2   BEGIN
3
4     -- Variáveis
5     DECLARE v_finished INTEGER DEFAULT 0;
6     DECLARE v_desDatas TEXT;
7
8     -- Cursor
9     DECLARE cursor_data CURSOR FOR SELECT DATE_FORMAT(DATAHORA ADM, '%Y-%m-%d') FROM bd_urg.urg_inform_geral
10    UNION SELECT DATE_FORMAT(DATAHORA ALTA, '%Y-%m-%d') FROM bd_urg.urg_inform_geral
11    UNION SELECT DTA_NASCIMENTO FROM bd_urg.urg_inform_geral;
12
13    DECLARE CONTINUE HANDLER FOR NOT FOUND SET v_finished = 1;
14
15    OPEN cursor_data;
16
17    get_data: LOOP
18
19      FETCH cursor_data INTO v_desDatas;
20
21      IF v_finished = 1 THEN
22        LEAVE get_data;
23      END IF;
24
25      -- Preencher a tabela DIM_DATA
26      INSERT INTO DIM_DATA(data) VALUES (v_desDatas);
27      COMMIT;
28
29    END LOOP get_data;
30
31    CLOSE cursor_data;
32

```

Figura 4: Procedure ”PREENCHE\_DIM\_DATA()”

### 2.1.3 Tabela de Factos

Nesta secção optou-se por apresentar apenas um exemplo, visto que todos os factos são tratados de maneira semelhante.

Como é possível verificar na figura 5, começou-se *script* por inicializar as variáveis referentes às chaves estrangeiras de cada tabela de dimensão que será utilizada para guardar os dados finais.

```

/*----- Variáveis -----*/
DECLARE v_finished INTEGER DEFAULT 0;
DECLARE v_idUrgencia INT(11);
DECLARE v_tempo_adm_alta INT(11);
DECLARE v_idade_paciente INT(11);
DECLARE v_cod_local INT(11);
DECLARE v_cod_proveniencia INT(11);
DECLARE v_cod_data_adm INT(11);
DECLARE v_cod_data_alta INT(11);
DECLARE v_cod_data_nascimento INT(11);
DECLARE v_cod_especialidade INT(11);
DECLARE v_cod_causa INT(11);
DECLARE v_cod_genero INT(11);

```

Figura 5: Variável finais do povoamento da tabela de Factos

Posteriormente, o grupo decidiu declarar o resto das variáveis de cada tabela de dimensão de forma a auxiliar a utilização eficiente dos cursores. Na figura seguinte apresentamos um exemplo dessas mesmas variáveis pertence à tabela de dimensão Especialidade.

```
-- ESPECIALIDADE
DECLARE v_finished_especialidade INTEGER DEFAULT 0;
DECLARE cod_especialidade1 INT(11);
DECLARE especialidade1 TEXT;
DECLARE especialidade2 VARCHAR(250);
```

Figura 6: Variável tabela de dimensão para cursos

Após a declaração de todas as variáveis auxiliares para a implementação dos cursos, passamos a declaração destes para todas as tabelas de dimensão. Para todas as tabelas de dimensão foram selecionadas as chaves e os cursos foram ordenados pelas mesmas.

No exemplo seguintes apresentamos a declaração do cursor da Data e do cursor da Especialidade.

```
/*-- Declaração dos cursos --*/

-- DATA
DECLARE cursor_data CURSOR FOR
SELECT DISTINCT id_data
FROM DIM_DATA
ORDER BY id_data ASC;

-- ESPECIALIDADE
DECLARE cursor_especialidade CURSOR FOR
SELECT DISTINCT id_especialidade
FROM DIM_ESPECIALIDADE
ORDER BY id_especialidade ASC;
```

Figura 7: Declaração dos cursores da Data e da Especialidade

De seguida, tal como foi feito nas tabelas de dimensão, foi necessário desenvolver um LOOP para todos os cursos de maneira a preencher as variáveis finais com os respetivos valores na tabela de factos. Como é possível observar com o exemplo da Figura 8, este LOOP irá percorrer toda a base de dados "BD\_URG" à procura dos dados pedidos, recolhendo-os para os guardas nas variáveis finais, funcionando como um ciclo externo.

```

DECLARE CONTINUE HANDLER FOR NOT FOUND SET v_finished = 1;

OPEN cursor_idUrgencia;

get_idUrgencia: LOOP

SET v_finished = 0;

FETCH cursor_idUrgencia INTO v_idUrgencia, v_tempo_adm_alta, v_idade_paciente;

IF v_finished = 1 THEN
    LEAVE get_idUrgencia;
END IF;

```

Figura 8: Ciclo externo

De modo a complementar estes ciclos externos, são definidos vários ciclos internos para cada cursor de cada tabela de dimensão com o objetivo de verificar o identificador que está a ser tratado naquele momento no LOOP exterior. É importante referir que este processo é feito em cada iteração para cada tabela de dimensão.

Na figura seguinte pode-se ver um exemplo do que foi mencionado referente à Especialidade.

```

-- ESPECIALIDADE
SELECT ALTA_DES_ESPECIALIDADE INTO especialidade1 FROM bd_urg.urg_inform_geral WHERE URG_EPISODIO=v_idUrgencia;
SET v_finished_especialidade = 0;
OPEN cursor_especialidade;
get_especialidade: LOOP
FETCH cursor_especialidade INTO cod_especialidade1;
IF v_finished_especialidade = 1 THEN
    LEAVE get_especialidade;
END IF;
-- Atribuição de v_cod_especialidade
SELECT descricao_esp INTO especialidade2 FROM DIM_ESPECIALIDADE WHERE id_especialidade = cod_especialidade1;
IF (especialidade1 LIKE especialidade2) THEN
    SET v_cod_especialidade = cod_especialidade1;
    SET v_finished_especialidade = 1;
END IF;
END LOOP get_especialidade;
CLOSE cursor_especialidade;

```

Figura 9: Ciclo interno

Por fim, é necessário preencher a tabela de factos com as variáveis finais e, para tal, estas são adicionadas à tabela e é finalizado o ciclo criado quando não existir mais nenhum dado da "BD\_URG" por analisar.

```
-- Preenche a tabela de factos FACTS_URG
INSERT INTO FACTS_URG(idUrg, id_proveniencia, id_admissao, id_alta,
id_nascimento, id_especialidade, id_causa, id_genero, id_local,
tempo_adm_alta, idade_paciente)
VALUES (v_idUrgencia, v_cod_proveniencia, v_cod_data_adm, v_cod_data_alta,
v_cod_data_nascimento, v_cod_especialidade, v_cod_causa, v_cod_genero,
v_cod_local, v_tempo_adm_alta, v_idade_paciente);
COMMIT;

END LOOP get_idUrgencia;

CLOSE cursor_idUrgencia;
```

Figura 10: Atribuição das variáveis finais

## 2.2 Talend

Nesta secção recorremos ao Talend para realizar a tarefa de ETL, este apresenta uma interface mais intuitiva. Tendo sido disponibilizado um ficheiro no formato ”.csv”, ”**urg\_inform\_geral.csv**”, respetivo ao *dataset* sobre o qual nos foi indicado trabalhar.

A informação do *dataset* referia 1000 ocorrências e disponibilizava dados relacionados com Urgências Hospitalares.

Deste modo desenvolvemos começamos a criar conexões as respectivas bases de dados, as querries que viriam a ser necessárias, bem como os parâmetros de uma *Job* de *Talend* que fosse capaz de povoar as Tabelas conforme desejado.

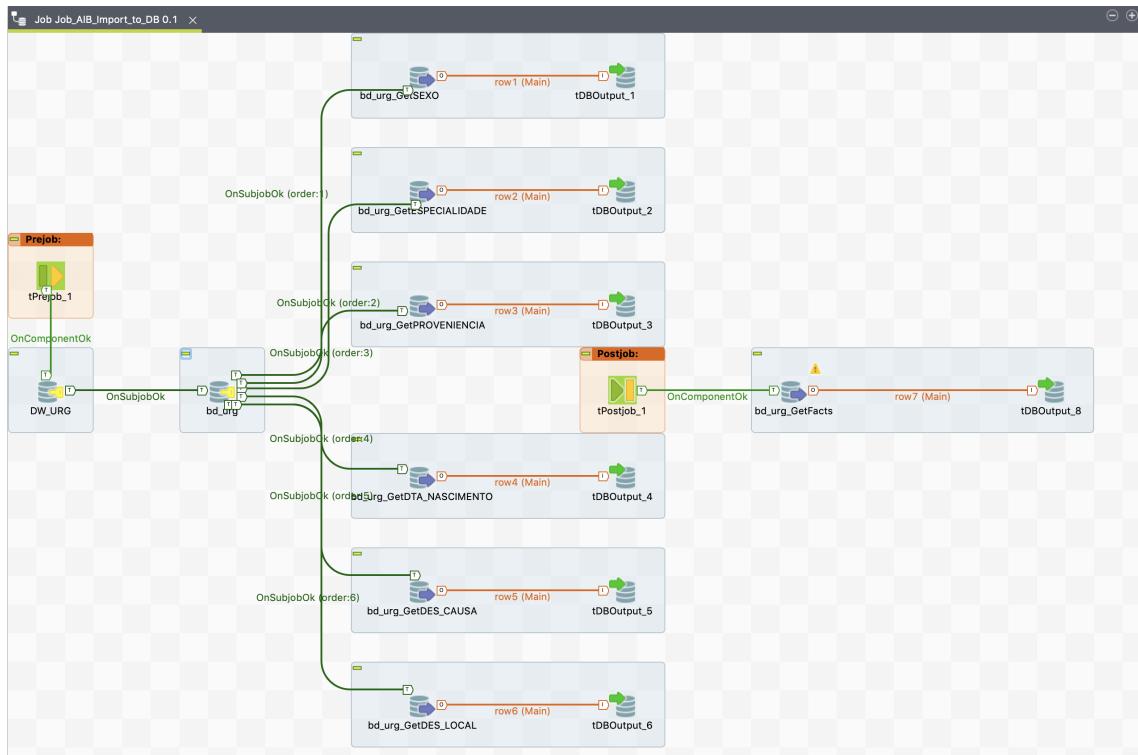


Figura 11: *Job*

### 2.2.1 Povoamento

Para que pudéssemos iniciar o tratamento dos dados começamos por criar as respetivas bases de dados, DW\_URG (o nosso *datawarehouse*) onde iríamos guardar a informação, ainda por povoar. E também criamos a bd\_urg já povoada com a informação do ficheiro **"urg\_inform\_geral.csv"**.

Tendo as Bases de dados e as Conexões configuradas começamos então o processo de elaboração já Job de Povoamento. Esta começa por se ligar as bases de dados e, existindo ligação, começa a povoar todas as tabelas de dimensão a partir dos dados de bd\_urg.

### 2.2.2 Conexão às Bases de Dados

Existindo as bases de dados numa instância de MySQL Workbench criamos as conexões respectivas a cada uma para poder ligar o Talend de modo a permitir trabalhar sobre as mesmas.

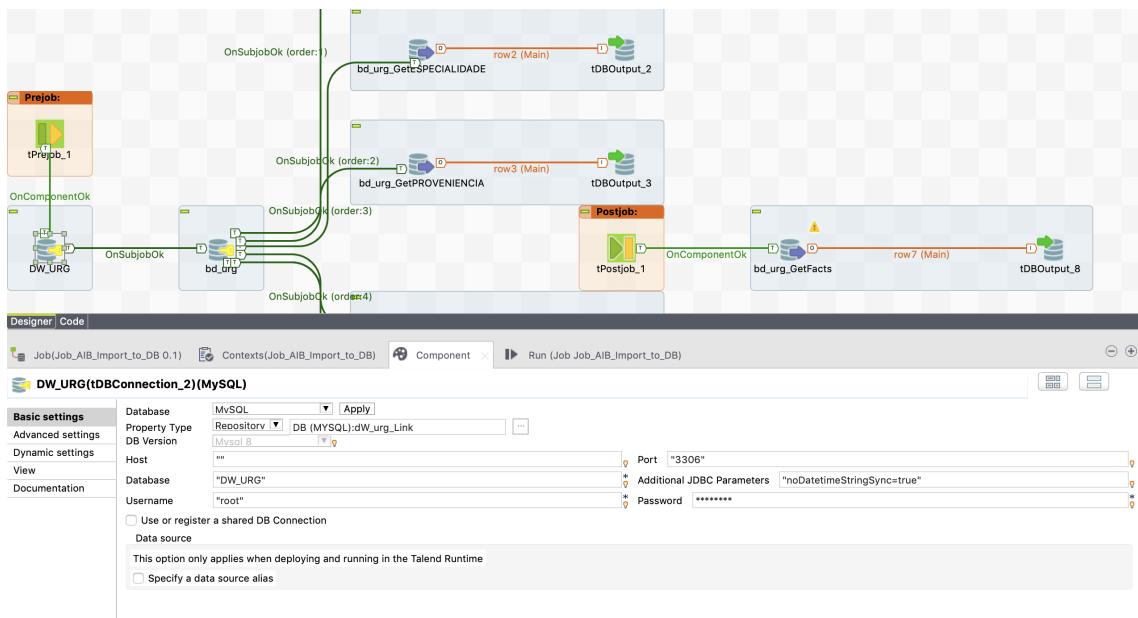


Figura 12: Ligação a Base de Dados dentro da Job

### 2.2.3 Tabelas de Dimensão

As tabelas de Dimensão são povoadas todas do mesmo modo. Recorremos a uma query que pesquisa nas respectivas colunas por diferentes valores dando-os como resposta.

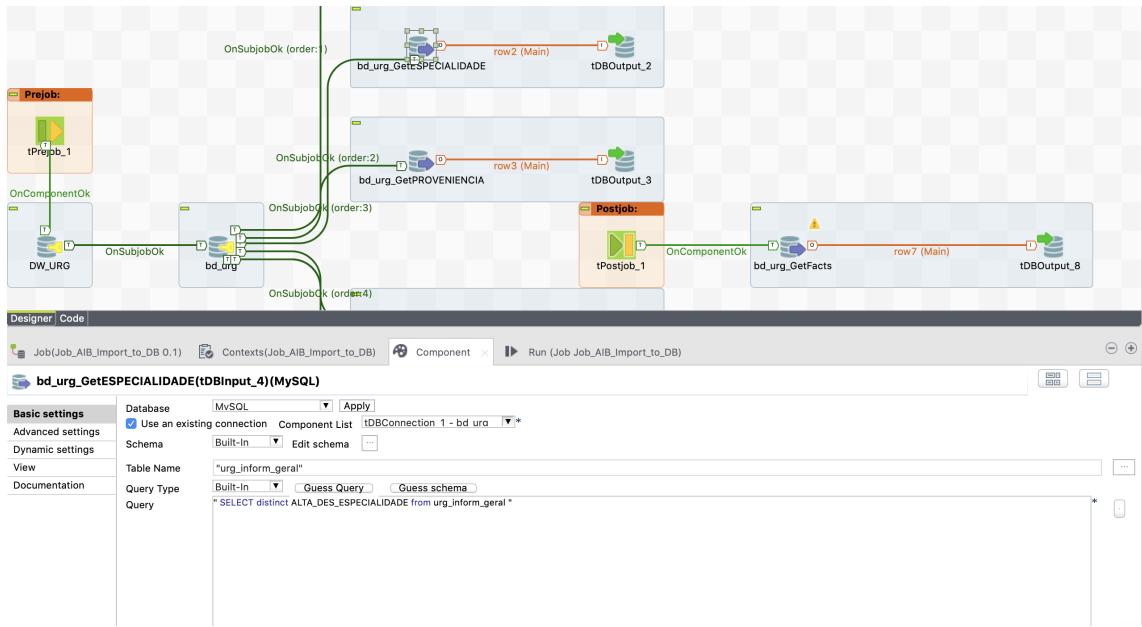


Figura 13: Recolha da informação para o Talend da base de dados bd\_urg.

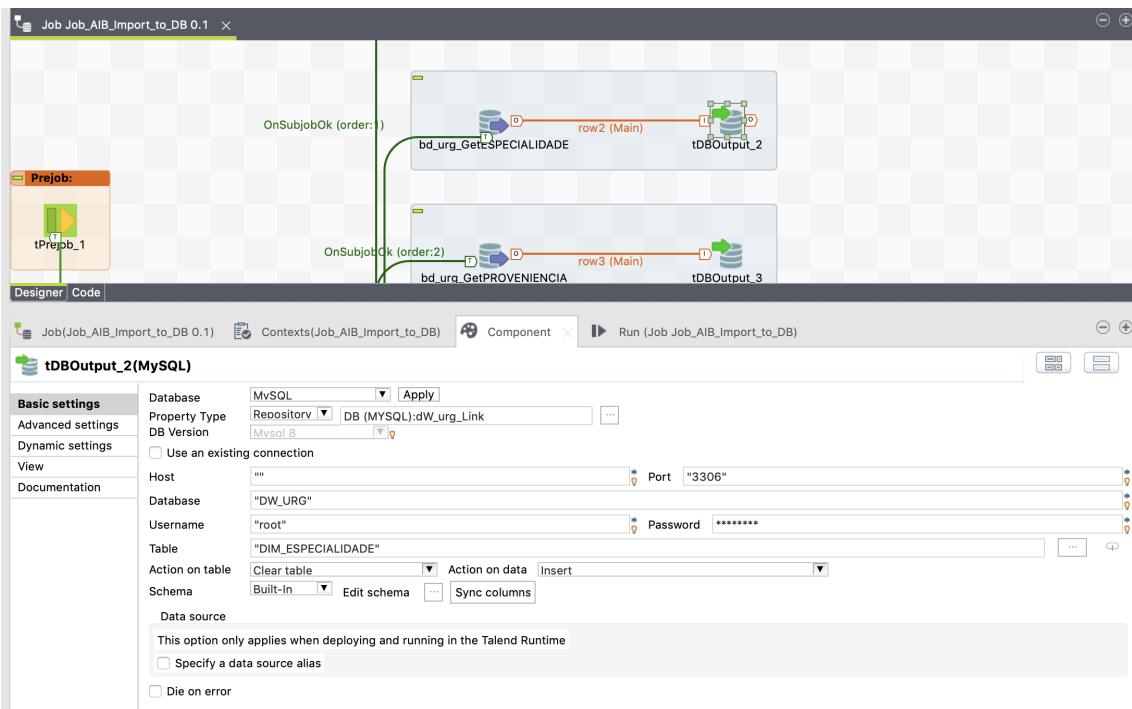


Figura 14: Inserção dos dados obtidos na Tabela de Dimensão respetiva.

#### 2.2.4 Tabelas de Factos

Apenas após terminar todo o povoamento das tabelas de dimensão e que se inicia o *PostJob* no qual a partir de bd\_urg e das tabelas de dimensão povoadas no *PreJob* procedemos a povoar a tabela de factos com toda a informação necessária.

Para tal utilizamos uma query que definimos que faz a coorelação entre os dados de cada entrada e o seu respetivo Id nas tabelas de dimensões.

```
SELECT    URG_EPISODIO as idURG,
          (select id_especialidade from DW_URG.DIM_ESPECIALIDADE
           where descricao_esp = bd_urg.urg_inform_geral.ALTA DES_ESPECIALIDADE)
          as id_especialidade,
          (select id_proveniencia from DW_URG.DIM_PROVENIENCIA
           where descricao_prov = bd_urg.urg_inform_geral.DES_PROVENIENCIA)
          as id_proveniencia,
          (select id_causa from DW_URG.DIM_CAUSA
           where descricao_cau = bd_urg.urg_inform_geral.DES_CAUSA)
          as id_causa,
          (select id_local from DW_URG.DIM_LOCAL
           where descricao_loc = bd_urg.urg_inform_geral.DES_LOCAL)
          as id_local,
          (select id_genero from DW_URG.DIM_GENERO
           where descricao_gen = bd_urg.urg_inform_geral.SEXO)
          as id_genero,
          (select id_data from DW_URG.DIM_DATA
           where data = str_to_date(bd_urg.urg_inform_geral.DTA_NASCIMENTO, '%Y-%m-%d'))
          as id_nascimento,
          (select id_data from DW_URG.DIM_DATA
           where data = str_to_date(bd_urg.urg_inform_geral.DATAHORA_ADM, '%Y-%m-%d'))
          as id_admissao,
          (select id_data from DW_URG.DIM_DATA
           where data = str_to_date(bd_urg.urg_inform_geral.DATAHORA_ALTA, '%Y-%m-%d'))
          as id_alta,
          TIMESTAMPDIFF(YEAR,DTA_NASCIMENTO,DATAHORA_ADM) AS idade_paciente,
          TIMESTAMPDIFF(MINUTE,DATAHORA_ADM,DATAHORA_ALTA) AS tempo_adm_alta
FROM bd_urg.urg_inform_geral
ORDER BY URG_EPISODIO;
```

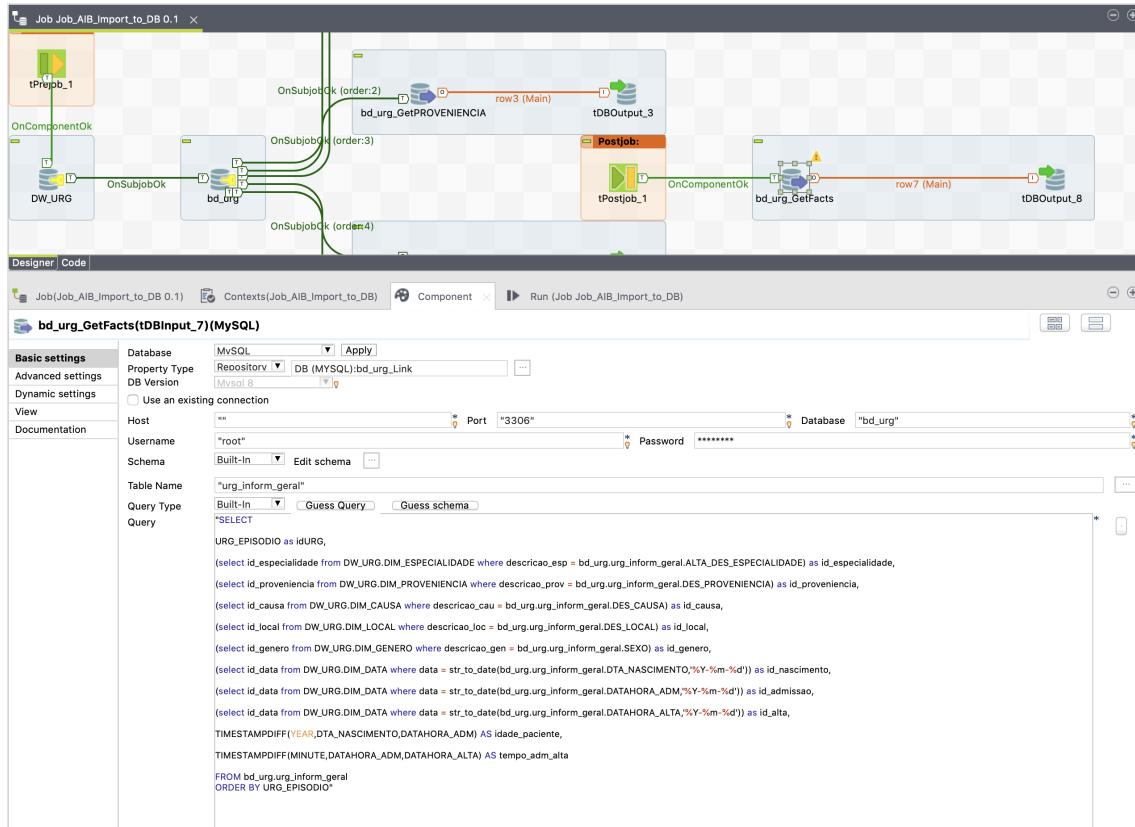


Figura 15: Recolha da informação para o Talend da base de dados bd\_urg.

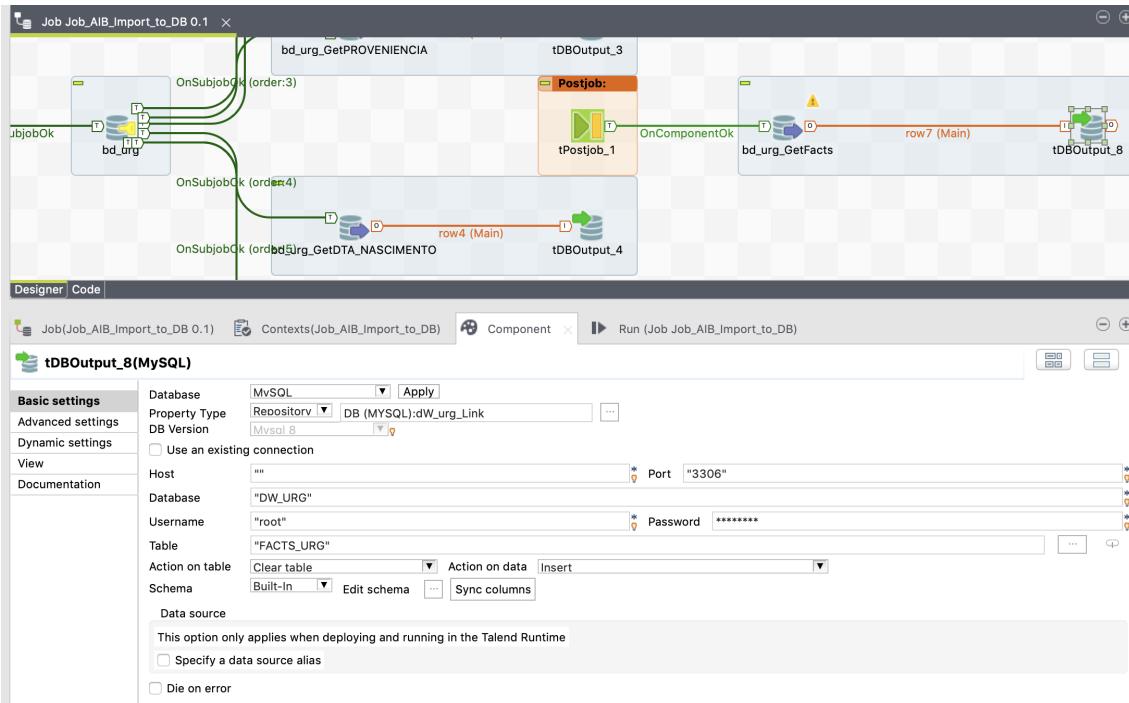


Figura 16: DB\_Output\_Facts

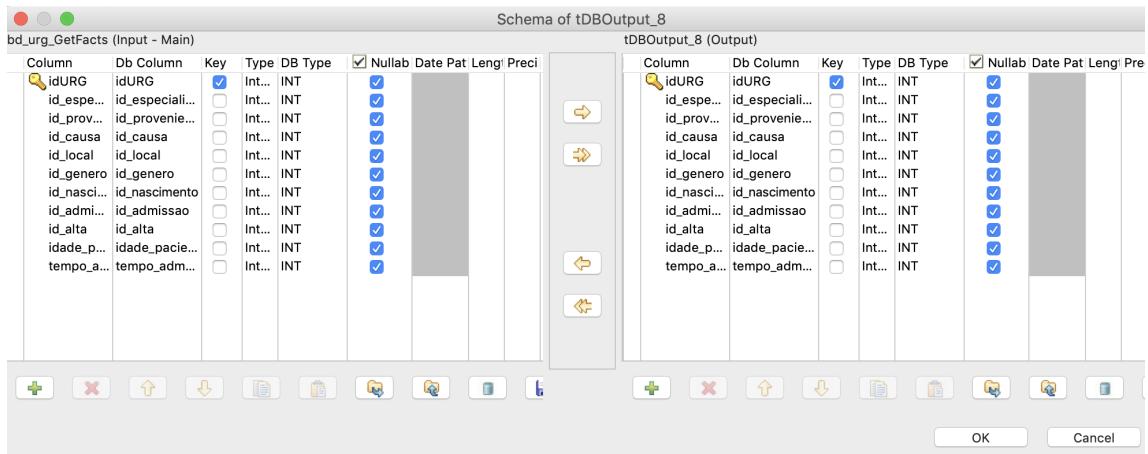


Figura 17: Output Schema

### **2.3 MySQL vs Talend**

Os métodos de trabalho em MySQL e Talend para o processo de ETL e tratamento de dados para o povoamento de Tabelas de Dimensão e Factos, são bastante distintos, não só no seu planeamento como utilização.

A nosso ver, a plataforma do Talend destaca-se pela sua interface bastante intuitiva que o torna uma excelente proposta para alguém que inicie a sua experiência no processo ETL, isto é para alguém sem muito conhecimento de MySQL, o Talend é capaz de permitir um processo ETL simples mas bem estruturado capaz de atingir os objetivos pretendidos.

No entanto se nos quisermos focar em tratamento de dados, temos de admitir que com o MySQL tivemos muita mais manobra de acção no que toca à manipulação e conversão de dados tal que conseguimos atingir uma DataWarehouse mais complexa.

No âmbito deste trabalho, tendo em conta a dimensão, tema do assunto e quantidade de atributos, o grupo acredita que a utilização do Talend é a mais acertada, não invalidando as capacidades do MySQL com o qual também conseguimos a criação do DataWarehouse.

## 2.4 Indicadores Clínicos *POWER BI*

Nesta secção foi necessário utilizar o POWER BI com os dados que foram criados no *MySQL* de maneira a que fosse possível desenvolver os indicadores clínicos. Assim sendo, realizou-se uma conexão com a base de dados *MySQL* desenvolvida, ou seja, a base de dados "DW\_URG", através das ferramentas disponibilizadas pelo POWER BI. Após de fornecer os dados referentes à base de dados (servidor e nome), foi possível importar todas as tabelas criadas anteriormente, como se pode verificar na figura seguinte.

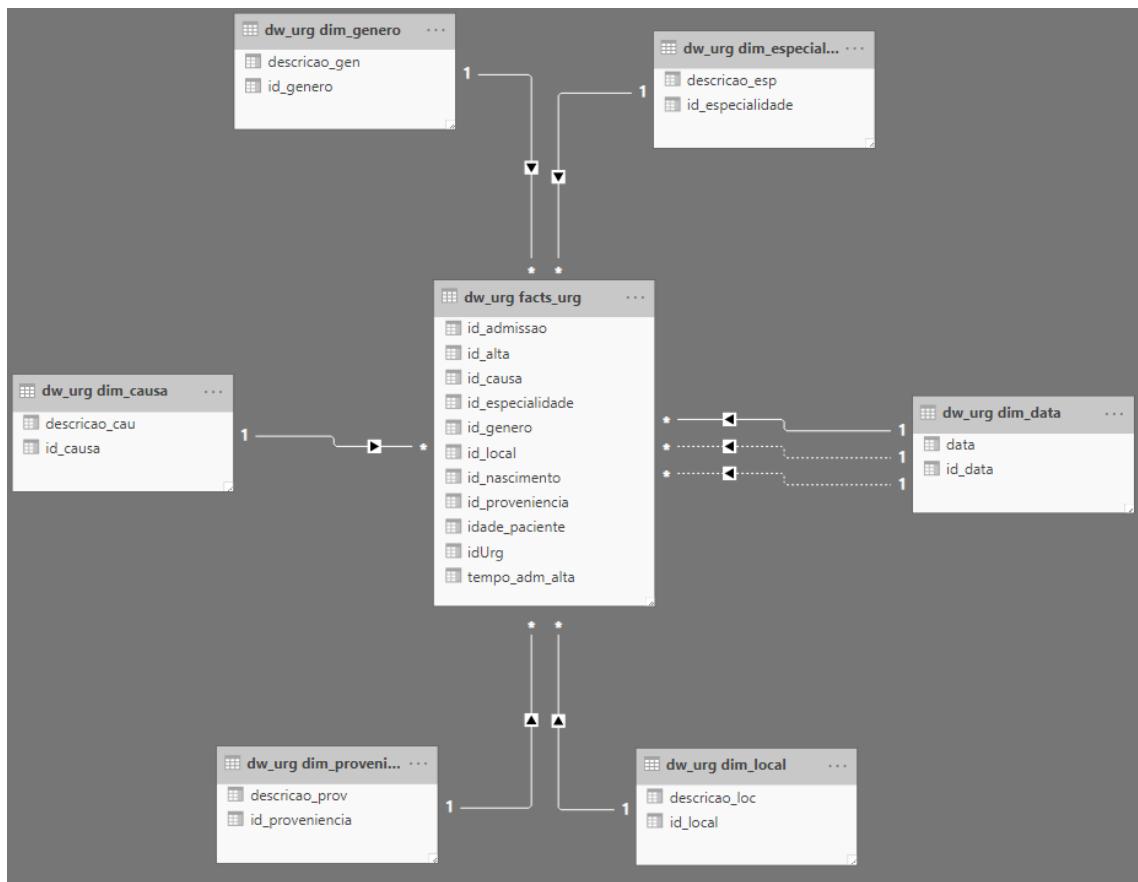


Figura 18: Base de dados no POWER BI

Feita a conexão da base de dados com o POWER BI, iniciou-se o desenvolvimento dos indicadores clínicos que serão apresentados a seguir.

#### 2.4.1 Média de tempo de alta do Paciente por Causa

Decidiu-se iniciar o desenvolvimento dos indicadores clínicos através da média de tempo que um paciente demora a ter alta mediante a causa. Como tal, este indicador apresenta num gráfico circular as diferentes causas e a média associada, ou seja, é possível visualizar no gráfico as sete causa diferentes, acidente de trabalho, acidente de viação, acidente escolar, acidente pessoal, doença, queda e outras e as suas respetivas médias.

Verificamos que as causas que têm maior média de tempo são os acidentes de viação (id 2) e de trabalho (id 1).

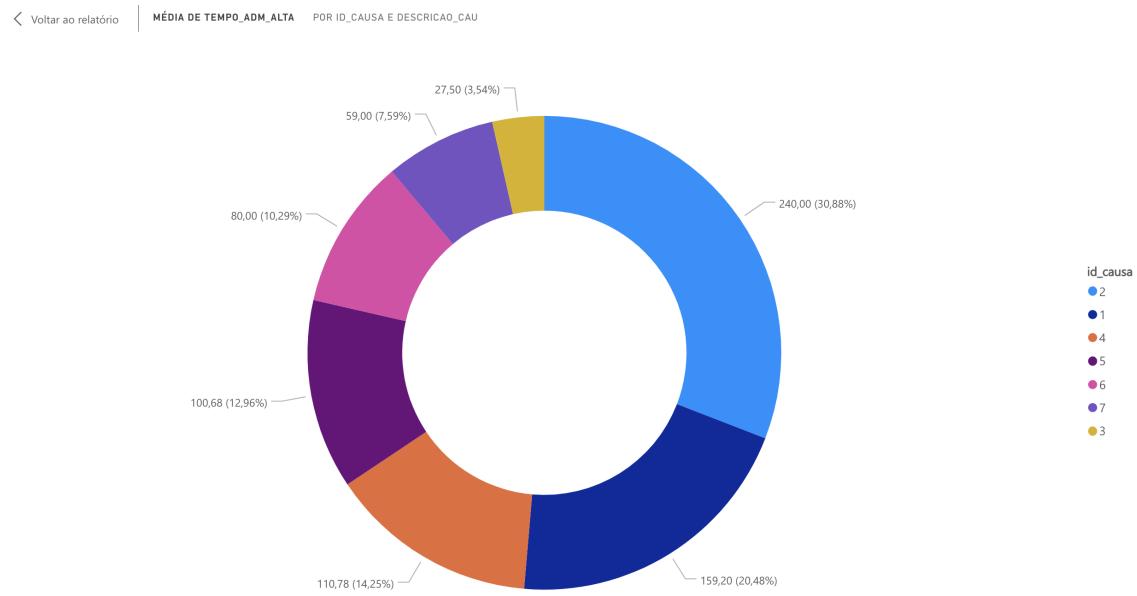


Figura 19: Média de tempo de alta do Paciente por Causa

#### 2.4.2 Média de tempo de alta do Paciente por Género

De seguida, optamos por desenvolver um indicador semelhante ao anterior, onde calculamos a média do tempo de alta do paciente por género, masculino e feminino. Como é possível analisar, os dois géneros têm um tempo de alta bastante semelhante, como género feminino a ter um tempo ligeiramente menor que o género masculino.

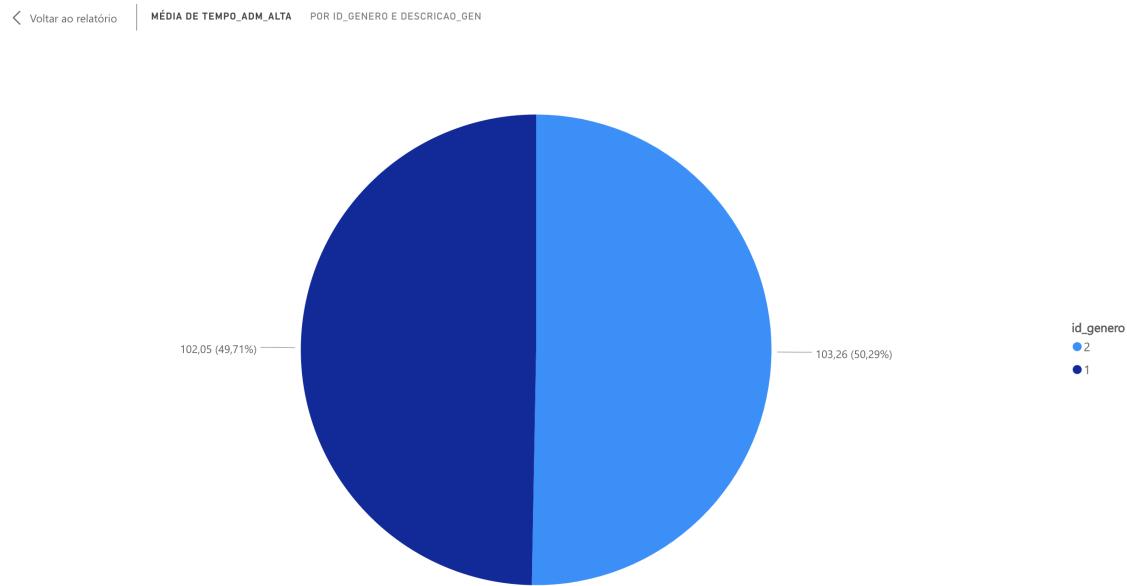


Figura 20: Média de tempo de alta do Paciente por Género

#### 2.4.3 Contagem por Género

Em relação a este indicador, pode-se dizer que corresponde a uma contagem por género, isto é, é apresentado neste indicador o número de pessoas por género, a média de tempo de alta por género (à semelhança do anterior) e ainda a média de idade de cada género. Analisando os dados verifica-se que existem mais mulheres neste conjunto de dados e que a sua idade média é superior à dos homens, apesar do seu tempo de admissão a alta ser inferior.

descricao_gen	id_genero	Contagem de id_genero	Média de tempo_adm_alta	Média de idade_paciente
F	1	555	102,05	46,98
M	2	445	103,26	42,02
<b>Total</b>		<b>1000</b>	<b>102,59</b>	<b>44,78</b>

Figura 21: Contagem por Género

#### 2.4.4 Idade do Paciente por Causa

Relativamente a este indicador, o objetivo foi apresentar uma estatística do máximo da idade dos pacientes nas diferentes causas de urgência. Conforme se pode visualizar pelo gráfico, as idades variam entre os 15 os 95 anos. A idade menor corresponde a acidentes escolares com a idade máxima nos 15 anos, já os acidentes com as maiores idades dizem respeito a doenças e acidentes pessoais com as idades máximas de 95 e 93, respetivamente.

De referir ainda que os acidentes de trabalho, com o registo máximo de idade de 63 anos, é normal que não ultrapasse os 66 anos, visto que é a idade de reforma em Portugal. Por fim, a idade máxima referente aos acidentes de viação neste conjunto de dados é de 50 anos, o que está em coerência com o resto dos dados, visto que a maioria deste tipo de acidentes acontece entre os 20 e os 60 anos de idade.

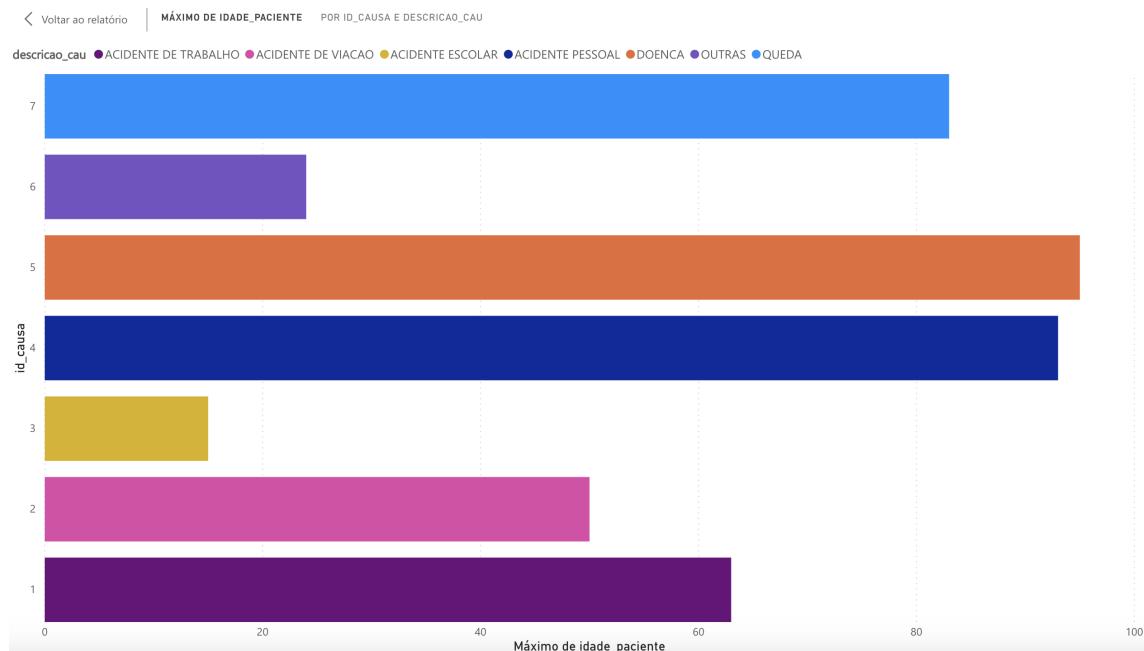


Figura 22: Idade do Paciente por Causa

#### 2.4.5 Média de idade do Paciente por Causa

Para além da idade máxima, decidiu-se criar este indicador clínico para apresentar também a idade média dos pacientes por causa de urgência. Desta forma, podemos observar no gráfico circular da figura seguinte que a maioria da média de idades encontra-se no intervalo dos 40 aos 50 anos. Curiosamente, a média de idade dos acidentes escolares é de 15 anos, ou seja, igual à idade máxima dos acidentes escolares registados.

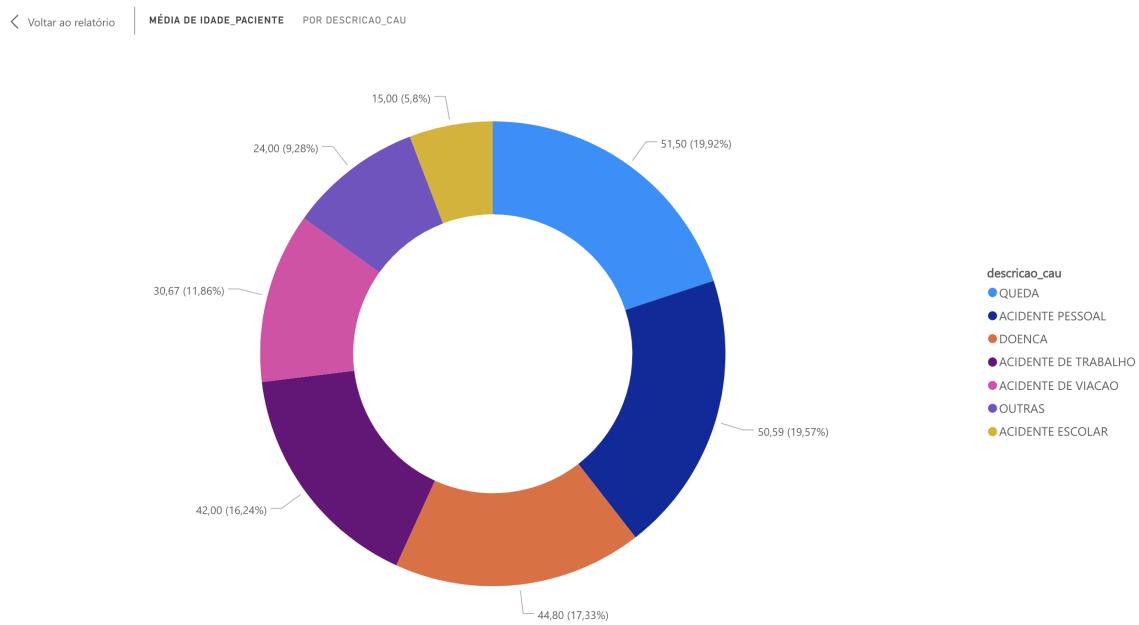


Figura 23: Média de idade do Paciente por Causa

#### 2.4.6 Ocorrências de Causas

Outro indicador que o grupo achou interessante realizar foi um indicador que mostrasse o número de ocorrências de urgência por tipo de causa. Para tal foi desenvolvido este gráfico que nos apresenta este número de ocorrências, onde se pode destacar as ocorrências por doença, que representam quase a globalidade das ocorrências (940 casos).

Para além dos casos de doença, temos os acidentes pessoais e de trabalho com valores semelhantes entre eles, 27 e 25, respectivamente. Por último temos o resto dos casos que correspondem a valores residuais.

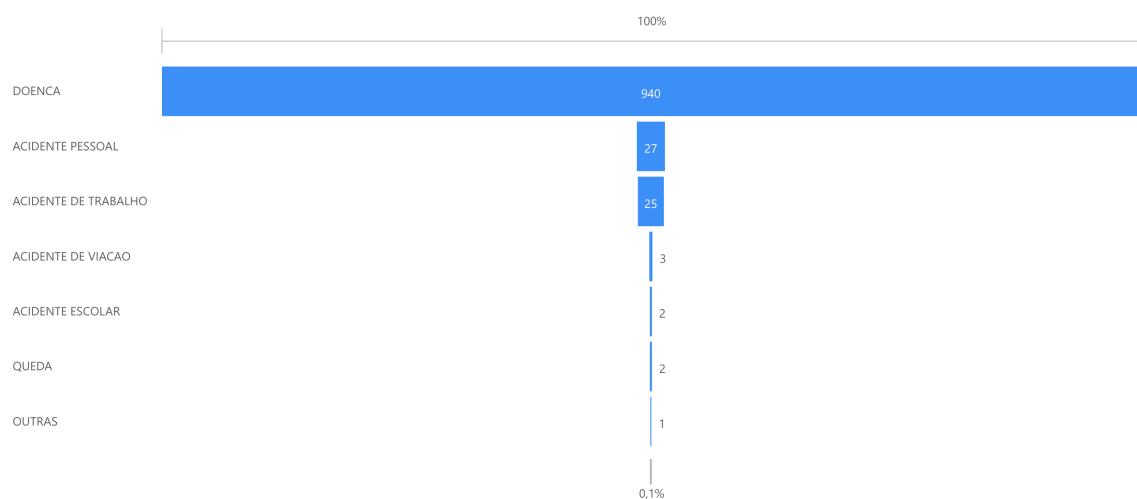


Figura 24: Ocorrências de Causas

#### 2.4.7 Género por Causa

Como último indicador, elaborou-se um gráfico com a quantidade de pacientes, masculinos e femininos, por tipo de urgência. Este indicador ser como complemento do anterior que apenas difere do anterior na introdução do género do paciente. Mais uma vez temos a doença em destaque face aos outros tipo de urgência.

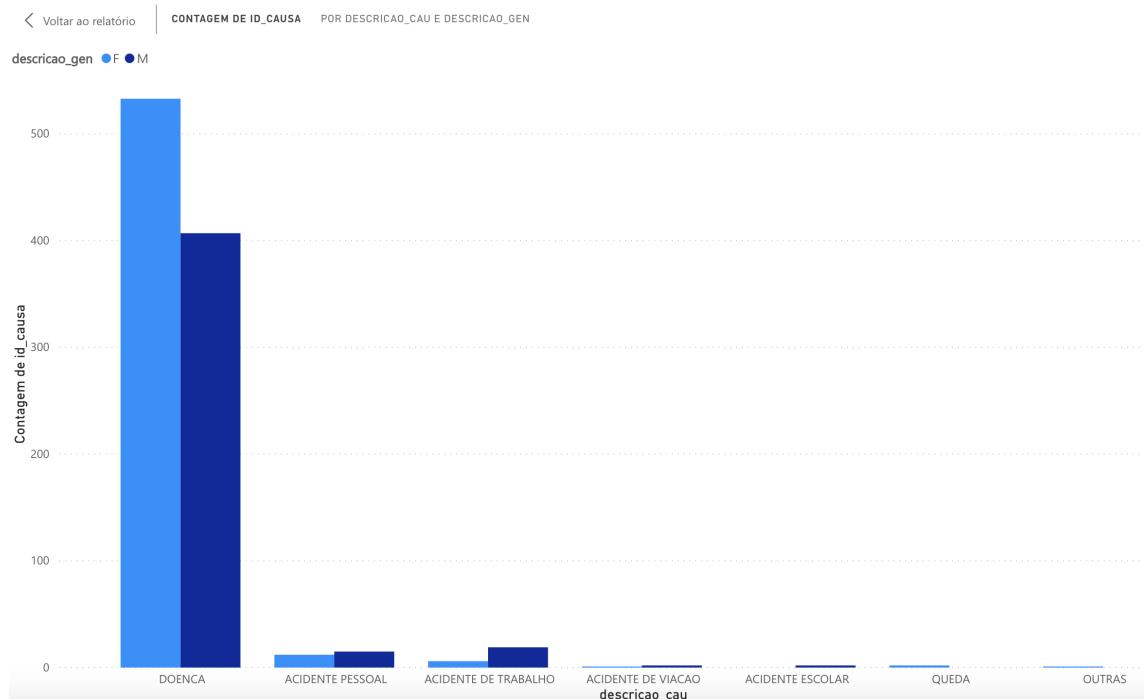


Figura 25: Género por Causa

## 2.5 Dashboard

Por fim, após definirmos todos os indicadores clínicos que o grupo achou serem mais relevantes, construíram-se duas *Dashboards* onde se colocaram os indicadores que, por base, têm características mais semelhantes entre eles.

### 2.5.1 Dashboard: "Causas"

A *Dashboard* "Causas" é constituída por indicadores clínicos que, acima de qualquer outro aspecto, apresentam informações estatísticas relativas às diferentes causas de urgências do hospital, permitindo ainda fazer análises de afluência por causa, por género, tempos de espera, entre outros.

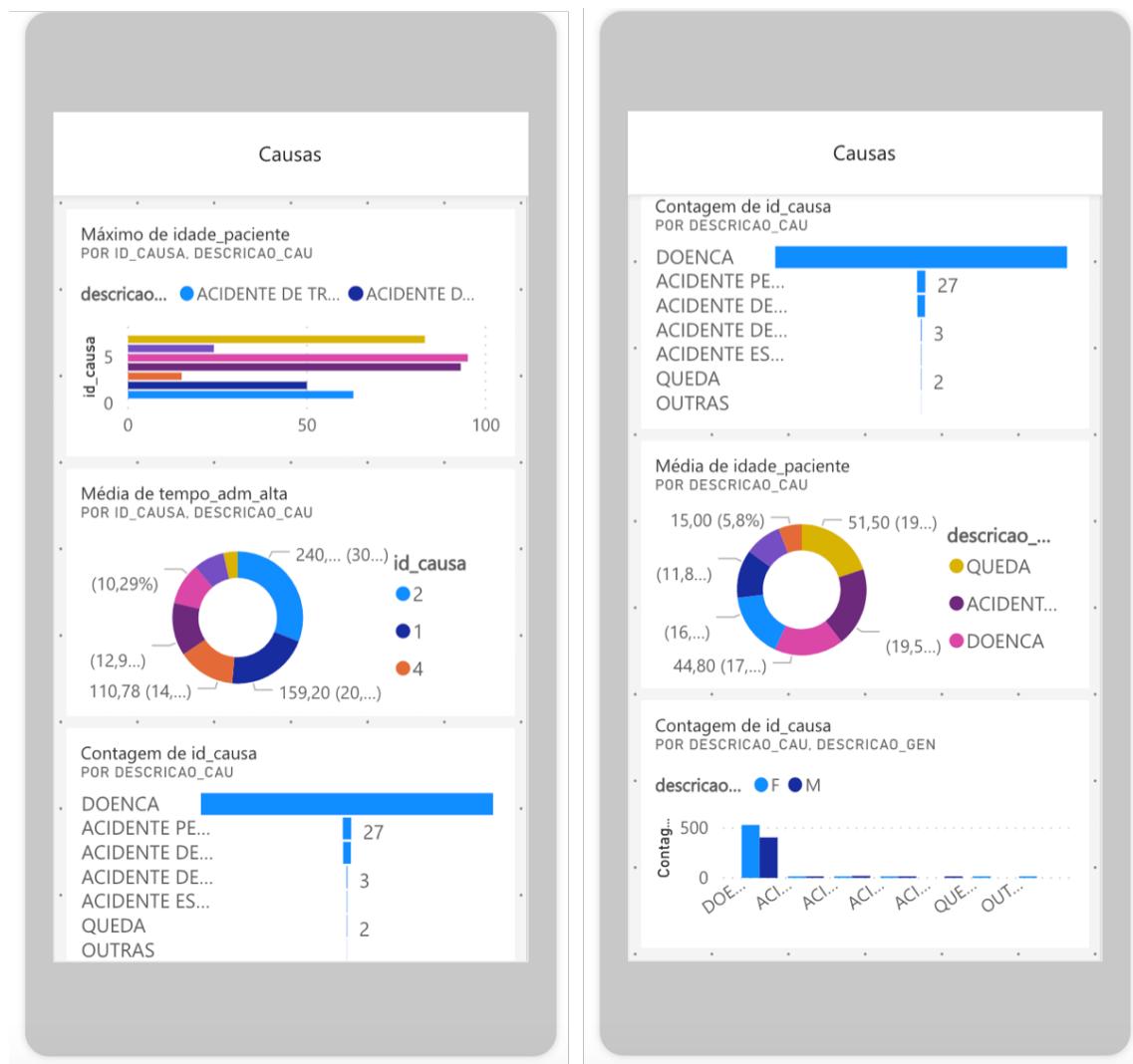


Figura 26: *Dashboard: "Causas"*

### 2.5.2 Dashboard: "Géneros"

Já o *Dashboard* "Géneros" permite ao utilizador formar uma ideia estatística relativa aos diferentes dados em relação ao sexo masculino e sexo feminino, tais como médias de tempo de admissão a alta, total de pacientes de cada género, pacientes de cada género por causa, entre outros dados estatísticos.

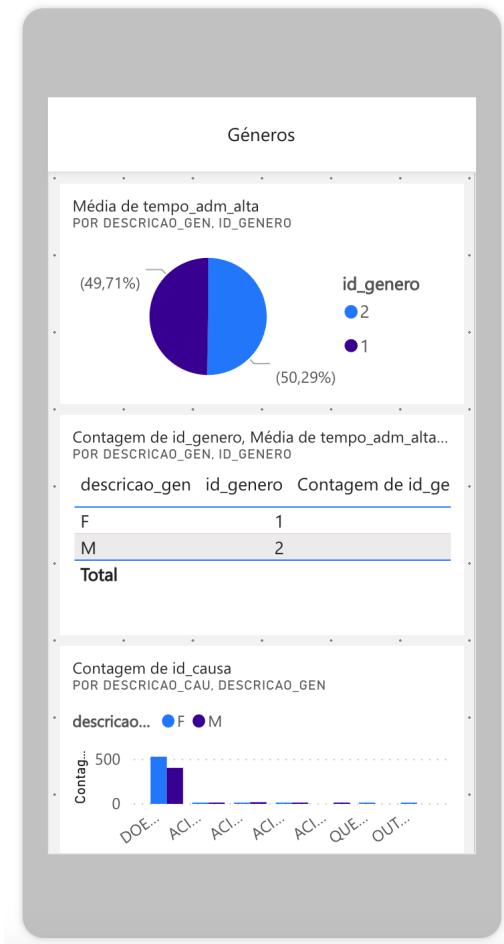


Figura 27: *Dashboard: "Géneros"*

## 2.6 Mockups

Nesta secção decidiu-se apresentar dois exemplos de como seria possível apresentar os indicadores desenvolvidos, tanto numa plataforma web como numa plataforma móvel, aos utilizadores do programa de gestão de urgências num hospital nacional.

### 2.6.1 Aplicações Web

Numa plataforma web, de maneira a tornar a utilização intuitiva, achamos que o programa deveria dar ao utilizador uma opção de escolha dos atributos que quer utilizar para a criação dos seus dados estatísticos bem como as diferentes formas de apresentação. Espera-se também que o software seja capaz de normalizar os dados e gerar o output esperado, tal como é apresentado no exemplo a seguir.

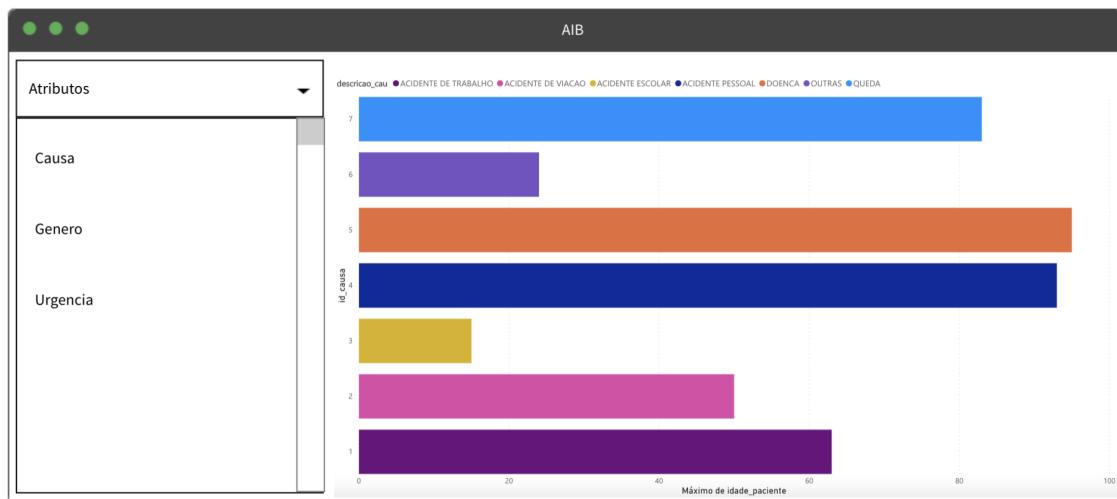
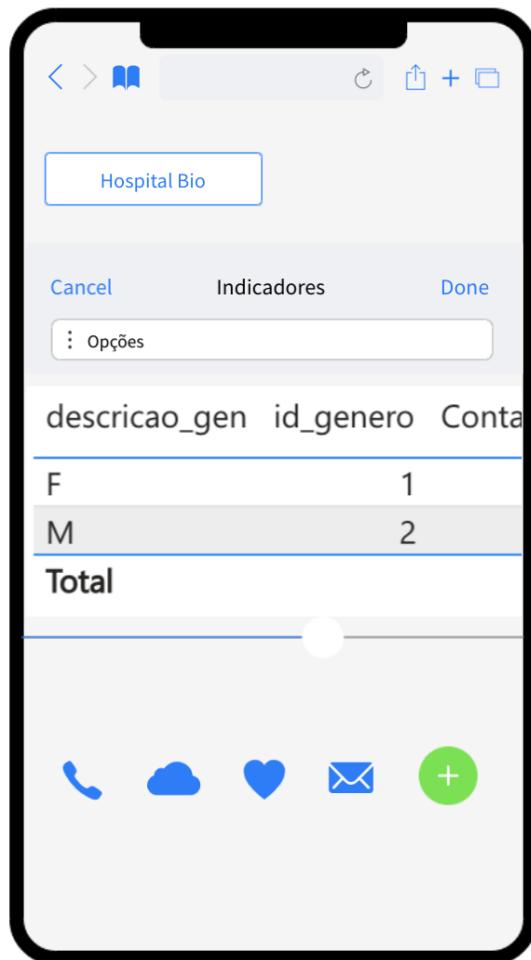


Figura 28: Género por Causa

### 2.6.2 Aplicações Móveis

Já numa vertente mais móvel, gostaríamos que a filosofia fosse a mesma que a anterior, onde o Utilizador tivesse as mesmas funcionalidades que dispõe no seu computador e ainda que tivesse atalhos para mais funcionalidades como é mostrado na figura seguinte. Achamos que para além do tratamento de dados, da apresentação dos *dashboard*, também é essencial a partilha dos mesmos.



### 3 Conclusões

Acabado o projeto há que fazer uma retrospectiva de todo o trabalho efetuado. Com este trabalho foi evidente a necessidade de nós termos conhecimento nas ferramentas para desenvolver um *data warehouse* que achamos interessante para um futuro profissional.

Além disso, ferramenta como **Talend Open Studio**, em que consegue extrair informações de modo a ser analisado foi muito importante, além do já estudado, nos anos anteriores do **MySql**. Informação esta que pode ser analisada, estudada através da plataforma **PowerBI**.

Em relação a esta última ferramenta que achamos muito poderosa para estudar os dados que nos são fornecidos, sentimos algumas dificuldades iniciais para fazer alguns dos gráficos, ainda que a ferramenta seja intuitiva. Passadas essas dificuldades iniciais, tudo o resto foi feito com alguma tranquilidade.

Em relação ao **Talend**, e depois de nos ocorrerem erros ao construir a **job** que devia-se à incompatibilidade entre o Sistema Operativo e o programa, durante algum tempo não conseguimos fazer a conexão entre este mesmo programa e o **MySql**. Ultrapassando esse problema, tudo o resto foi feito ainda que tenhamos enfrentado algumas dificuldades para a elaboração dos requisitos propostos.

Concluímos então este trabalho com a afirmação de que este trabalho todo efetuado desde Setembro foi muito importante para os nossos conhecimentos nesta área que achamos interessante.