

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

APLICAÇÕES INFORMÁTICAS NA BIOMEDICINA

---

## Ficha 5

Bernardo Silva (a77230)  
César Ribeiro (a79014)  
Mariana Fernandes (a81728)  
Paulo Jorge (a78203)  
Raul Vilas Boas (a79617)

---

18 de Novembro de 2019

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Jobs</b>	<b>2</b>
2.1	Job 1 . . . . .	2
2.2	Job 2 . . . . .	3
2.2.1	Map e criação da tabela em MySQL . . . . .	4
2.3	Job 3 . . . . .	4
<b>3</b>	<b>Conclusão</b>	<b>5</b>

# 1 Introdução

Este relatório tem como finalidade expor a proposta de resolução que o grupo realizou para responder à ficha 5.

A ficha possui em anexo um ficheiro csv denominado “mental\_health” que contém informação correspondente a aspetos pessoais e profissionais, e um questionário que avalia a presença de perturbações mentais nos profissionais das áreas TI. O ficheiro possui 21 atributos definidos.

O objetivo deste trabalho é então definir, criar e executar diversas *jobs* utilizando o programa *Talend*.

## 2 Jobs

### 2.1 Job 1

No primeiro *job* vamos filtrar os trabalhadores que procuraram tratamento para uma doença mental. Isto é, vamos pegar nos dados fornecidos no *mental\_health.csv* e filtrar apenas os dados dos quais na coluna respectiva ao *treatment* aparece a string “Yes”. O *output* vai ser direccionado para um novo ficheiro excel com todas as informações dos trabalhadores que procuraram tratamento.,

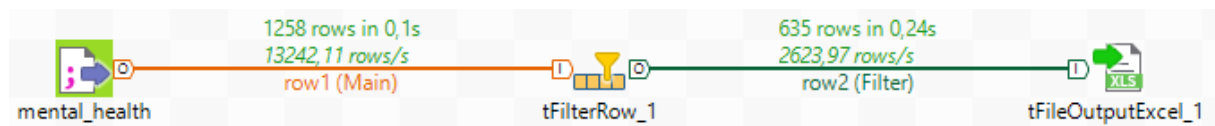


Fig. 1: Esquema do *job1* no Talend.

Na figura 2 está definido o filtro usado no *tFilterRow* que vai ter influenciar o resultado.

Input column	Function	Operator	Value
treatment	Match	==	"Yes"

Fig. 2: Filtro definido no *tFilterRow*

Na tabela 1 está representado uma pequena parte dos resultados obtidos ao correr o primeiro *job*. Nesta tabela conseguimos comprovar que na coluna *treatment* todas as linhas possuem um “Yes”.

Tabela 1: Resultados parciais do *job 1*

Timestamp	Age	Gender	Country	State	Self Employed	Family History	Treatment
2014-08-27 11:29:31	37	Female	United States	IL	NA	No	Yes
2014-08-27 11:29:46	31	Male	United Kingdom	NA	NA	Yes	Yes
2014-08-27 11:31:50	35	Female	United States	MI	NA	Yes	Yes
2014-08-27 11:32:39	42	Female	United States	IL	NA	Yes	Yes
2014-08-27 11:32:44	31	Male	United States	OH	NA	No	Yes
2014-08-27 11:33:23	42	female	United States	CA	NA	Yes	Yes

## 2.2 Job 2

Para o Job 2 definimos que o objetivo era explorar a facilidade de dar baixa médica numa empresa de grandes dimensões. As colunas essenciais para este *job* são *no\_employees* e *leave*. Primeiro começamos por criar um conexão à base de dados para posteriormente podermos inserir os dados na base de dados. De seguida usamos um *tMap* para inserir os dados do *mental\_health* na base de dados. Na figura 3 está representado o esquema do *job2* no Talend.

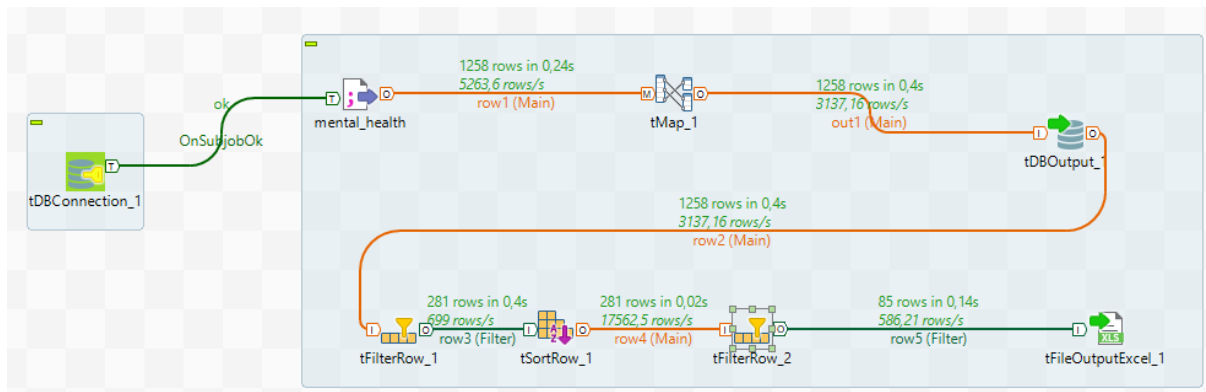


Fig. 3: Esquema do *job2* no Talend.

Depois disto começamos a filtrar a informação. O primeiro filtro usado consistiu em filtrar apenas as empresas que tenham mais de 1000 trabalhadores e cujas idades sejam maiores que 0, devido ao facto de alguns dados terem idades inválidas. O segundo serve para filtrar apenas os trabalhadores que consideram fácil ter baixa médica para problemas de saúde mental.

Conditions	Input column	Function	Operator	Value
	no_employees	Empty	==	"More than 1000"
	Age	Empty	>=	0
Conditions	Input column	Function	Operator	Value
	leave	Empty	==	"Somewhat easy"
	leave	Empty	==	"Very easy"
	leave	Empty	==	"Yes"

Fig. 4: Filtros definidos no *tFilterRow*

Por último, ordenamos por ordem crescente a idade e guardamos o resultado num ficheiro *excel*.

Schema column	sort num or alpha?	Order asc or desc?
Age	num	asc

Fig. 5: Parâmetro definido no *tSortRow*

Na tabela 2 está representado uma pequena parte dos resultados obtidos do *job2*. Assim conseguimos comprovar que a idade está ordenada por ordem crescente e que as colunas *no\_employees* e *leave* foram filtradas corretamente.

Tabela 2: Resultados parciais do *job2*

Timestamp	Age	Gender	(...)	no_employees	(...)	leave
2014-08-28 05:05:32	20	F	(...)	More than 1000	(...)	Somewhat easy
2014-08-27 14:27:51	22	Female	(...)	More than 1000	(...)	Somewhat easy
2014-08-27 15:15:42	22	F	(...)	More than 1000	(...)	Somewhat easy
2014-08-28 01:38:53	23	female	(...)	More than 1000	(...)	Somewhat easy
2014-08-28 17:20:41	23	Female	(...)	More than 1000	(...)	Somewhat easy
2014-08-27 15:36:13	24	Female	(...)	More than 1000	(...)	Somewhat easy

### 2.2.1 Map e criação da tabela em MySQL

Para a criação da tabela MySQL, utilizamos um map para, a cada coluna, adaptar os dados para estar em conformidade com o MySQL.

(Segue-se a seguir uma imagem descritiva)

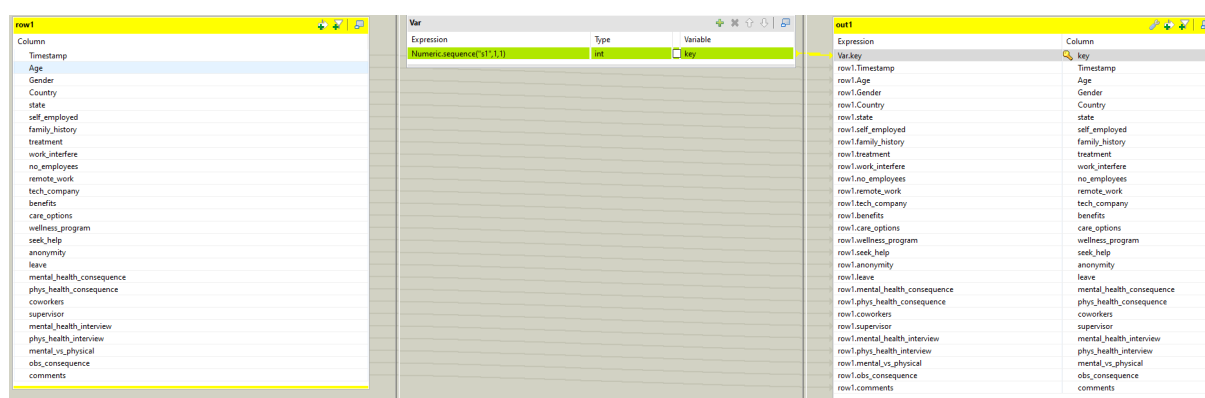


Fig. 6: Definições avançadas do processo *tMap*

Como os dados fornecidos não possuem nenhum campo que serviria como primary key, precisamos de criar a nossa própria primary key. Para isso, criamos uma nova coluna chamada "key" e seleccionamos a checkbox "Key", como podemos ver na tabela da direita, na imagem acima. A seguir, criamos uma expressão que simplesmente incrementa de 1 em 1, a partir do número 1, para cada row (observar tabela do meio da imagem acima).

## 2.3 Job 3

Para o Job 3 vamos verificar de um modo geral se pessoas que não se conformem com o género tem uma maior prevalência de doenças mentais na família do que em pessoas que se conformam.

Na figura 7 está representado o esquema do resultado no *Talend*.

Primeiro filtramos os trabalhadores que responderam como M,m, Male, F, f, Female e female ou "cis" e vamos guardar esse resultado num ficheiro. O resultado que não foi filtrado vai ser guardado noutro ficheiro

Para além disto, ordenou-se por *family\_history* para facilitar a observação do número de pessoas que responderam 'sim' nesta coluna. Dos resultados obtidos, cerca de metade dos não conformados verificou-se problemas de doenças mentais na família. Dos conformados apenas um terço tinha familiares com problemas mentais.

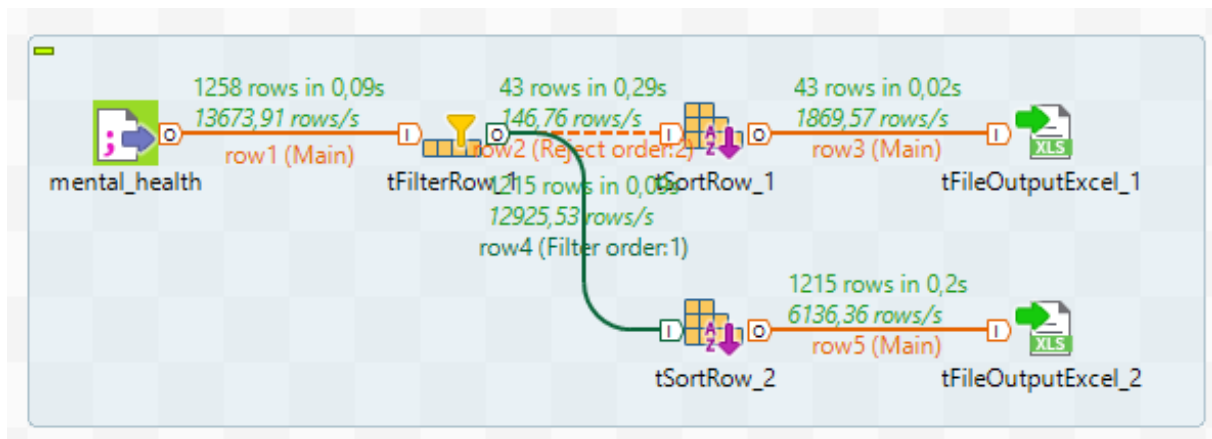


Fig. 7: Esquema do *job3* no Talend.

Conditions	Input column	Function	Operator	Value
	Gender	Empty	==	"M"
	Gender	Empty	==	"m"
	Gender	Empty	==	"Male"
	Gender	Empty	==	"male"
	Gender	Empty	==	"F"
	Gender	Empty	==	"f"
	Gender	Empty	==	"Female"
	Gender	Empty	==	"female"

☒ Use advanced mode

**Advanced**

```
// code sample : use input_row to define the condition.
// input_row.columnName1.equals("foo") ||!(input_row.columnName2.equals("bar"))
// replace the following expression by your own filter condition
(input_row.Gender.contains("cis"))||(input_row.Gender.contains("Cis"))||(input_row.Gender.contains("CIS"))
```

Fig. 8

### 3 Conclusão

A resolução desta ficha permitiu-nos utilizar a ferramenta *Talend* em paralelo com *MySql*, a ferramenta em si é complexa e requer experiência e por isso esta ficha foi uma mais valia para nos familiarizarmos com a mesma. Para concluir, o grupo acha que concretizou com sucesso as três *jobs* pedidas no enunciado.