

Instituto Politécnico do Cávado e do Ave

Integração de Sistemas de Informação

## ***Big Basket* – Processos ETL**

Licenciatura em Engenharia de Sistemas  
Informáticos

Nathaniel Joshua Armando da Silva Lloyd Jones  
Nº 21116

Barcelos, Portugal  
15/novembro/2022

# Conteúdo

Lista de Figuras .....	3
1. Introdução .....	4
1.1. Contextualização.....	4
1.2. Objetivos .....	4
1.3. Estrutura do documento.....	5
2. Estado de Arte .....	6
2.1. Extract, Transform and Load (ETL) Tools .....	6
3. Conteúdo do ficheiro .csv.....	7
3.1. <i>Pentaho Data Integration (PDI)</i> .....	7
4. Realização do trabalho.....	8
4.1. Transformações .....	9
4.2. <i>Jobs</i> .....	19
5. Conclusão .....	25
6. Referências.....	25

## Lista de Figuras

Figura 1 - Pequena Amostra do Ficheiro .csv .....	7
Figura 2 - Ficheiros .ktr .....	9
Figura 3 – TransformationSortItemCategory .....	9
Figura 4 - CSV file input .....	9
Figura 5 - Operações do CSV file input .....	10
Figura 6 - Sort rows by Category .....	10
Figura 7 - Operações Sort rows by Category .....	10
Figura 8 - Switch/ case.....	11
Figura 9 - Operação Switch/ case .....	11
Figura 10 - Gestão das Categorias .....	12
Figura 11 - Processamento de informação da categoria Baby Care .....	12
Figura 12 - JSON Baby Care.....	13
Figura 13 - Obtenção de campos do Step JSON Baby Care .....	13
Figura 14 - XML Baby Care.....	14
Figura 15 - Obtenção de campos do Step XML Baby Care .....	14
Figura 16 - Conexão à base de dados do MongoDB.....	14
Figura 17 - Ligação do output à coleção Baby Care .....	15
Figura 18 - Conexão a uma implantação do MongoDB.....	15
Figura 19 - Base de dados Big Basket no MongoDB .....	16
Figura 20 - Dados armazenados da categoria Baby Care no MongoDB.....	16
Figura 21 – TransformationCheapestByCategory .....	17
Figura 22 - Sort Baby Care por preço ascendente.....	17
Figura 23 - Sample Row Baby Care mais barato.....	18
Figura 24 - Sample rows do primeiro selecionado .....	18
Figura 25 - XML Cheapest.....	18
Figura 26 – TransformationMostExpensiveByCategory .....	18
Figura 27 - Job Send Email.....	19
Figura 28 – Start.....	19
Figura 29 - Transformação do SortItemCategory.....	20
Figura 30 - Mail validator.....	20
Figura 31 - Email de envio e destino .....	20
Figura 32 - Autenticação do SMTP Server e email em uso.....	21
Figura 33 - Email criado pelo Job.....	21
Figura 34 - Job HTML .....	22
Figura 35 - File exists XML ME (Most Expensive) .....	22
Figura 36 - File exists XML ME função .....	22
Figura 37 - Delete file HTML ME.....	23
Figura 38 - Delete file HTML ME função.....	23
Figura 39 - XSL Most_Expensive .....	23
Figura 40 - XSL Most_Expensive função .....	23
Figura 41 - HTML gerado pelo Job .....	24

# 1. Introdução

Neste capítulo será elaborada os conteúdos deste relatório, dando o leitor a perceber no que consiste este projeto, a contextualização e os objetivos do mesmo, tal como também a estrutura do documento.

## 1.1. Contextualização

Maior parte dos sistemas de informação exigem uma troca de dados, e com esta exigência podem aparecer vários problemas tais como uma sequência incorreta de informação, inconsistência dos dados ou informação errada ou até um formato de dados errado. Para evitar estes problemas surgiu a área de integração de dados, passando pela análise das regras e processos de negócio e, a partir daí, construir uma solução que permita a comunicação entre dois ou mais sistemas. Um dos processos de integração de sistemas passa pela Extração, Transformação e Carregamento (sigla ETL em inglês) dos dados.

A importância do ETL numa organização é diretamente proporcional quanto à organização do armazenamento de dados. As ferramentas de ETL coletam, leem e migram grandes volumes de dados brutos de várias fontes de dados e entre plataformas diferentes, carregando esses dados num único banco de dados ou um armazenamento de dados para facilitar o acesso. Após isso processam os dados para torná-los significativos com operações como classificação, junção, reformatação, filtragem, incorporação e agregação. Por fim, eles incluem interfaces gráficas para resultados mais rápidos e fáceis do que os métodos tradicionais de movimentação de dados por meio de pipelines de dados codificados manualmente.

As ferramentas modernas de ETL incluem inteligência integrada que deteta e reflete continuamente as alterações nos objetos de origem e destino para garantir a consistência dos dados – a força vital da estratégia e da experiência do cliente.

## 1.2. Objetivos

O principal propósito deste projeto é a transformação e extração de dados de um ficheiro *.csv*, chamado *Big Basket*, proveniente do website *Kaggle*.

Para a extração destes dados, deve resultar treze ficheiros *Extensible Markup Language (.xml)* e onze ficheiros *JavaScript Object Notation (.json)*.

### **1.3. Estrutura do documento**

O documento encontra-se organizado em 5 capítulos sendo estes a Introdução, Estado de Arte, Conteúdo dos Ficheiros, Realização do Trabalho, Conclusão e Referencias.

### **1.4. Link Video Youtube**

<https://youtu.be/uriO6MDWrog>

## 2. Estado de Arte

Qualquer supermercado, tanto de grandes dimensões como de pequenas, tem sempre uma vasta lista de artigos e quantidades dos mesmos, entre outras informações. Para a melhor gestão deste estabelecimento é crucial uma base de dados e uma tecnologia que manipule eficazmente o mesmo consoante as necessidades dos utilizadores.

Sem esta organização um funcionário terá de ver artigo a artigo o que tem em stock, sem mencionar os outros dados que tem de ser tomados em conta, tal como o prazo de validade. Com uma base de dados o trabalho deste funcionário será extremamente simplificado e com uma margem de erros no armazenamento dos dados quase inexistente.

Já existem programas que conseguem resolver estes problemas e foi ao estudar alguns deles que eu consegui desenvolver o meu projeto.

### 2.1. Extract, Transform and Load (ETL) Tools

As ferramentas *ETL* tratam de um processo de três passos de gestão de dados sendo o primeiro a extração dos dados de uma fonte de dados estruturada ou não estruturada, transformando-os num formato que satisfaça os requisitos funcionais e analíticos da empresa e, por fim, carrega os dados no alvo destino.

Um sistema de *ETL* bem desenvolvido extrai os dados, impõem padrões de qualidade dos dados, preenche os dados num formato padronizado para que varias fontes possam ser usadas juntas e entrega os dados prontos para que os desenvolvedores da aplicação a possam desenvolver e os donos do produto possam tomar decisões.

### 3. Conteúdo do ficheiro .csv

O ficheiro .csv tem os seguintes campos:

- *ProductName* do tipo *String* que guarda o nome do produto;
- *Brand* do tipo *String* que guarda a marca do produto;
- *Price* do tipo *Number* com formato “#.##” que guarda o preço do produto;
- *DiscountPrice* do tipo *Number* com formato “#.##” que guarda o preço do produto em desconto;
- *Quantity* do tipo *String* que guarda a quantidade do artigo em peças, litros ou kilos;
- *Category* do tipo *String* que guarda a categoria do produto;
- *SubCategory* do tipo *String* que guarda a subcategoria do produto;

```

ProductName,Brand,Price,DiscountPrice,Image_Url,Quantity,Category,SubCategory,Absolute_Url
Onion (Loose),Fresco,69.75,52.00,https://www.bigbasket.com/media/uploads/p/1/40075537_5-fresho-onion.jpg,2 kg,Fruits & Vegetables,"Potato, Onion & Tomato",
Onion (Loose),Fresco,174.35,130.00,https://www.bigbasket.com/media/uploads/p/1/1201414_1-fresho-onion.jpg,5 kg,Fruits & Vegetables,"Potato, Onion & Tomato",https://w
Onion (Loose),Fresco,34.87,26.00,https://www.bigbasket.com/media/uploads/p/1/10000148_30-fresho-onion.jpg,1 kg,Fruits & Vegetables,"Potato, Onion & Tomato"
Onion,Fresco,69.74,52.00,https://www.bigbasket.com/media/uploads/p/1/1201413_1-fresho-onion.jpg,2 kg,Fruits & Vegetables,"Potato, Onion & Tomato",https://w
Onion (Loose),Fresco,174.37,129.00,https://www.bigbasket.com/media/uploads/p/1/10000150_19-fresho-onion.jpg,5 kg,Fruits & Vegetables,"Potato, Onion & Tomat
"Farm Eggs - Table Tray, Medium, Antibiotic Residue-Free",Fresco,110.00,78.00,https://www.bigbasket.com/media/uploads/p/1/40072320_2-fresho-farm-eggs-table
"Farm Eggs - Table Tray, Medium, Antibiotic Residue-Free",Fresco,210.00,181.00,https://www.bigbasket.com/media/uploads/p/1/150502_6-fresho-farm-eggs-table-
Potato (Loose),Fresco,50.32,38.75,https://www.bigbasket.com/media/uploads/p/1/40048457_9-fresho-potato-new-crop.jpg,1 kg,Fruits & Vegetables,"Potato, Onion
Potato,Fresco,100.65,77.50,https://www.bigbasket.com/media/uploads/p/1/40162469_6-fresho-potato-new-crop.jpg,2 kg,Fruits & Vegetables,"Potato, Onion & Toma

```

Figura 1 - Pequena Amostra do Ficheiro .csv

#### 3.1. Pentaho Data Integration (PDI)

O *Pentaho Data Integration (PDI)* fornece os recursos *Extract, Transform e Load (ETL)* que facilitam o processo de captura, limpeza e armazenamento de dados usando um formato uniforme e consistente que é acessível e relevante para usuários finais e tecnologias *IoT (Internet of Things)*.

Os usos comuns do *Pentaho Data Integration* incluem:

- Migração de dados entre diferentes bases de dados e aplicações;
- Importação em massa de dados em bases de dados, aproveitando ao máximo os ambientes de processamento em *Cloud*, agrupamentos e ambientes de processamento paralelo maciços;
- Limpeza de dados com etapas que variam de transformações muito simples a muito complexas;
- Integração de dados, incluindo a capacidade de alavancar ETL em tempo real como fonte de dados para *Pentaho Reporting*;

- População de *data warehouse* com suporte integrado para dimensões de alteração lenta e criação de chave substituta.

## 4. Realização do trabalho

### **.CSV**

Para iniciar o trabalho foi preciso obter um ficheiro *.csv* (*Comma-seperated values*) adequado ao tema que é *stock* de um supermercado.

Um ficheiro *.csv* é caracterizado pela separação de informação por vírgulas, maior parte das vezes usado em bases de dados e folhas de cálculo. Estes tipos de ficheiros são usados para mover dados entre programas que não tem as capacidades básicas de trocas de informação.

### **.XML**

Um ficheiro *.xml* (*Extensible Markup Language*) é usado para estruturar dados para armazenamento e transporte.

Este ficheiro tem de ter compreensão fácil pois utilizam a língua do ser humano em vez de linguagem da máquina. A informação contida neles está etiquetada por categoria para que esta seja organizada e facilmente identificada.

### **.XSL**

Um ficheiro *.xsl* (*Extensible Stylesheet Language*) tem nele contigo código capaz de criar uma folha de dados que descreve os dados que serão apresentados na página *WEB*.

### **.HTML**

*HTML* (*Hyper Text Markup Language*) é a linguagem principal para o desenvolvimento de páginas *WEB*.



## 4.1. Transformações

As transformações estão guardadas num tipo de ficheiro *.ktr* (*Kettle Transformation File*). O meu projeto tem os seguintes ficheiros *.ktr*:

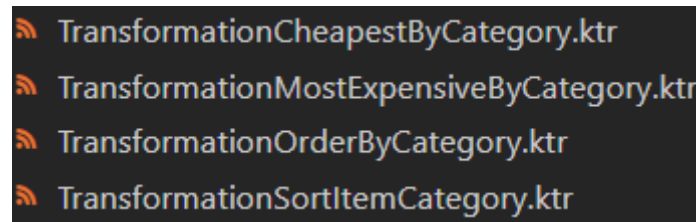


Figura 2 - Ficheiros *.ktr*

### *TransformationSortItemCategory*

A primeira transformação é a “*TransformationSortItemCategory*” que filtra as categorias dos artigos da base de dados e exporta os dados em *.json*, *.xml* e também para a base de dados *MongoDB*.

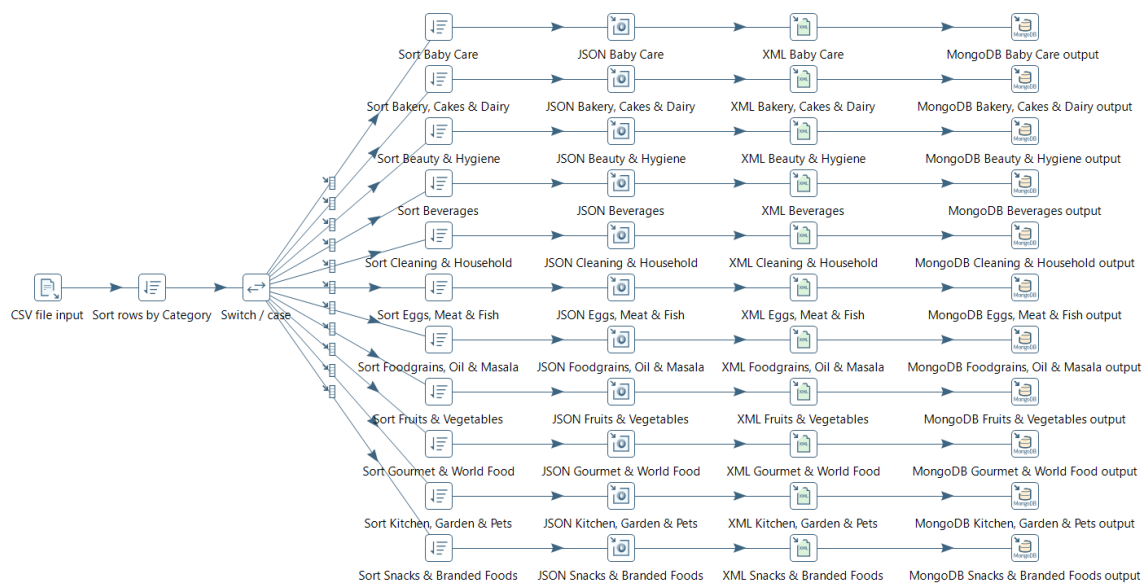


Figura 3 – *TransformationSortItemCategory*

Os dados são inseridos no *input* “*CSV file input*” que identifica as tabelas, nome, tipo de dados, formato e comprimento do ficheiro *.csv*.



Figura 4 - *CSV file input*

CSV file input

Step name: **CSV file input**

Filename: C:/Users/ossie/Desktop/ISI/Trabalho/BigBasket.csv Navega...

Delimiter: , Insert TAB

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result? ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	ProductName	String		152		€	,	.	none
2	Brand	String		10		€	,	.	none
3	Price	Number	##	15	0	€	.	,	none
4	DiscountPrice	Number	##	15	0	€	.	,	none
5	Quantity	String		10		€	,	.	none
6	Category	String		24		€	,	.	none
7	SubCategory	String		23		€	,	.	none

Help OK Obtem campos Preview Cancela

Figura 5 - Operações do CSV file input

Após a inserção dos dados, o campo com nome "Category" é filtrado por ordem alfabética no passo "Sort rows by Category".

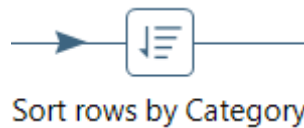


Figura 6 - Sort rows by Category

Sort rows

Nome do Step: **Sort rows by Category**

Sort directory: %%java.io.tmpdir%% Navega...

TMP-file prefix: out

Sort size (rows in memory): 1000000

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	Category	S	N	N	0	N

Figura 7 - Operações Sort rows by Category

Depois para organizar e separar as categorias recorri a um “Switch / case” que verifica o que está escrito dentro da coluna “Category” e compara o texto do formato *String* com o valor inserido no Switch/ case.

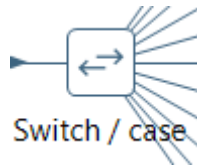


Figura 8 - Switch/ case

Switch / case

Step name: Switch / case

Field name to switch: Category

Use string contains comparison: ☒

Case value data type: String

Case value conversion mask:

Case value decimal symbol:

Case value grouping symbol:

Case values

#	Value	Target step
1	Baby Care	Sort Baby Care
2	Bakery, Cakes & Dairy	Sort Bakery, Cakes & Dairy
3	Beauty & Hygiene	Sort Beauty & Hygiene
4	Beverages	Sort Beverages
5	Cleaning & Household	Sort Cleaning & Household
6	Eggs, Meat & Fish	Sort Eggs, Meat & Fish
7	Foodgrains, Oil & Masala	Sort Foodgrains, Oil & Masala
8	Fruits & Vegetables	Sort Fruits & Vegetables
9	Gourmet & World Food	Sort Gourmet & World Food
1..	Kitchen, Garden & Pets	Sort Kitchen, Garden & Pets
1..	Snacks & Branded Foods	Sort Snacks & Branded Foods

Default target step:

Help OK Cancela

Figura 9 - Operação Switch/ case

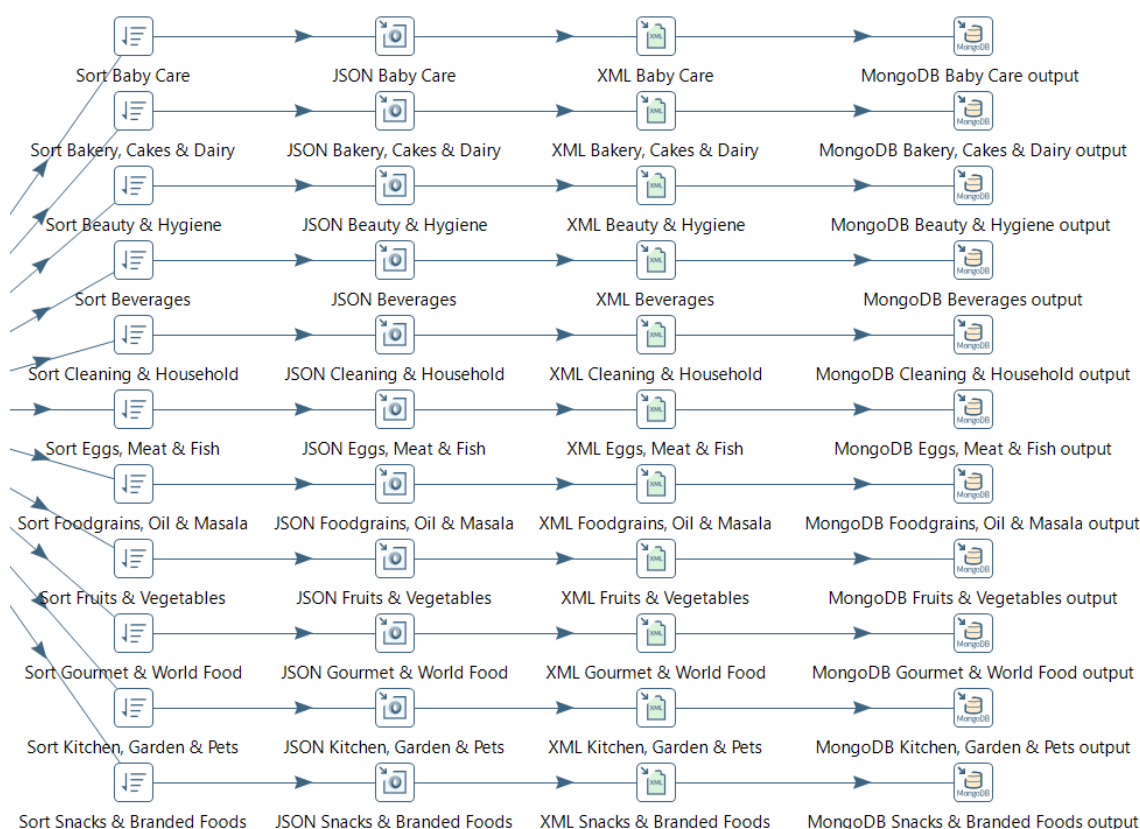


Figura 10 - Gestão das Categorias

### Gestão do *Baby Care*

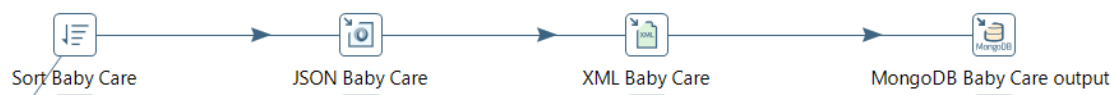


Figura 11 - Processamento de informação da categoria *Baby Care*

No início do processo é ligado ao *Switch/case* o *Sort Baby Care* que separa a categoria por *Baby Care* das restantes.

De seguida a categoria *Baby Care* é convertida para o formato *.json*, *.xml* e exportado para a base de dados no *MongoDB*.

## Step JSON Baby Care

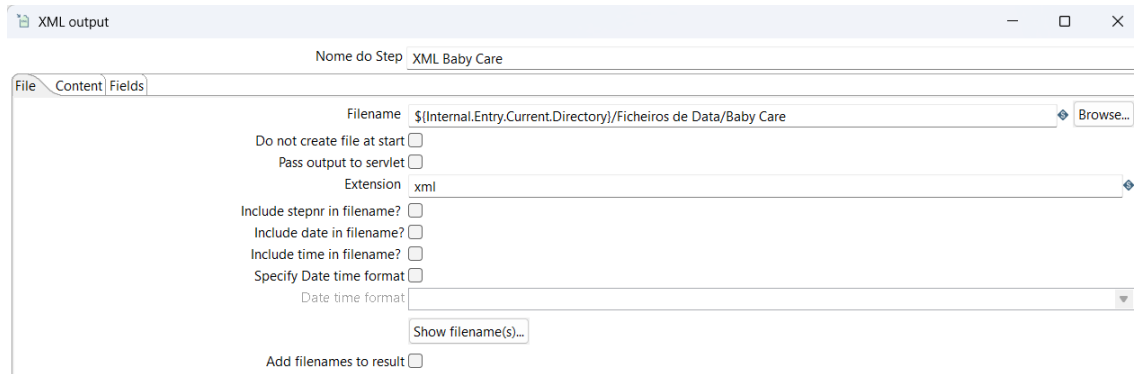
Figura 12 - JSON Baby Care

Aqui ocorre a conversão do formato .csv para .json e terá como diretório “\${Internal.Entry.Current.Directory}/Ficheiros de Data/Baby Care” sendo que o “\${Internal.Entry.Current.Directory}” servirá para encontrar sempre o diretório em que se encontra o ficheiro .csv.

#	Fieldname	Element name
1	ProductName	ProductName
2	Brand	Brand
3	Price	Price
4	DiscountPrice	DiscountPrice
5	Quantity	Quantity
6	Category	Category
7	SubCategory	SubCategory

Figura 13 - Obtenção de campos do Step JSON Baby Care

## Step XML Baby Care



XML output

Nome do Step XML Baby Care

File Content Fields

Filename \$(Internal.Entry.Current.Directory)/Ficheiros de Data/Baby Care Browse...

Do not create file at start ☐

Pass output to servlet ☐

Extension xml

Include stepnr in filename? ☐

Include date in filename? ☐

Include time in filename? ☐

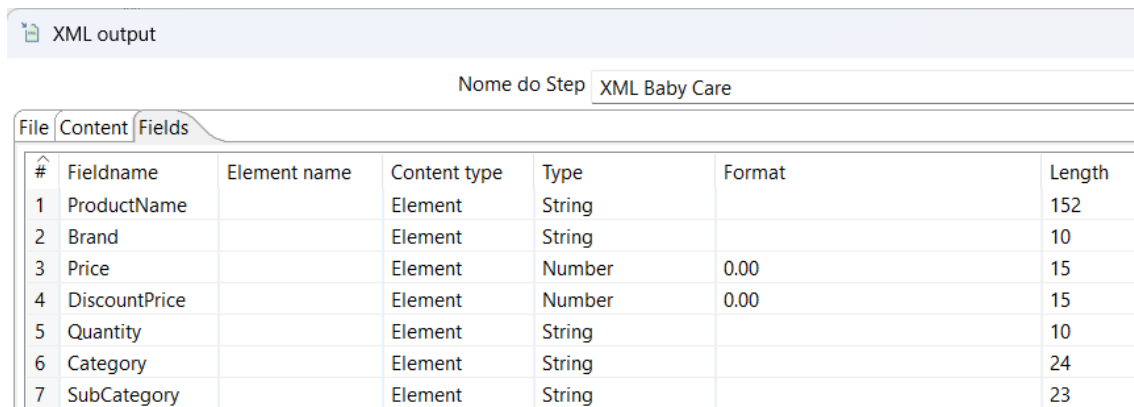
Specify Date time format ☐

Date time format

Show filename(s)...

Add filenames to result ☐

Figura 14 - XML Baby Care



XML output

Nome do Step XML Baby Care

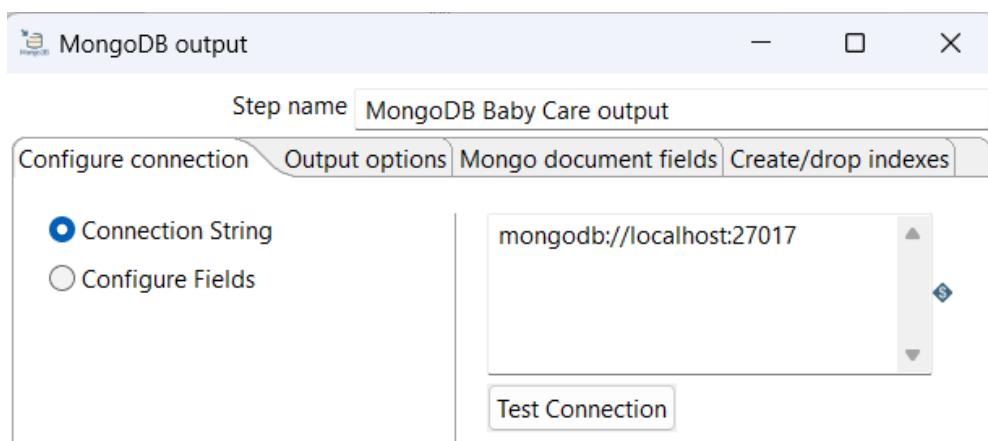
File Content Fields

#	Fieldname	Element name	Content type	Type	Format	Length
1	ProductName		Element	String		152
2	Brand		Element	String		10
3	Price		Element	Number	0.00	15
4	DiscountPrice		Element	Number	0.00	15
5	Quantity		Element	String		10
6	Category		Element	String		24
7	SubCategory		Element	String		23

Figura 15 - Obtenção de campos do Step XML Baby Care

Aqui ocorre a conversão do ficheiro .csv para .xml onde é guardado no ficheiro referido acima na figura 14 e a obtenção dos campos e respetivos tipos e formatos.

## Step MongoDB Baby Care output



MongoDB output

Step name MongoDB Baby Care output

Configure connection Output options Mongo document fields Create/drop indexes

☒ Connection String

☐ Configure Fields

mongodb://localhost:27017

Test Connection

Figura 16 - Conexão à base de dados do MongoDB

Step name: MongoDB Baby Care output

Configure connection | Output options | MongoDB document fields | Create/drop indexes

Database: BigBasket Get DBs

Collection: Baby Care Get collections

Batch insert size: 100

Truncate collection: ☐

Update: ☐

Upsert: ☐

Multi-update: ☐

Modifier update: ☐

Number of retries for write: 5

Figura 17 - Ligação do output à coleção Baby Care

Neste passo é necessário seleccionar a base de dados a que queremos ligar o output do MongoDB do nosso *PDI* e também a coleção onde queremos armazenar os campos e dados do *Baby Care*.

### Ligação à base de dados *MongoDB*

**New Connection**

Connect to a MongoDB deployment

FAVORITE

URI ? Edit Connection String

mongodb://localhost:27017

[Advanced Connection Options](#)

Save Save & Connect Connect

Figura 18 - Conexão a uma implantação do MongoDB

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
Baby Care	65.54 kB	1 K	222.00 B	1	45.06 kB
Bakery, Cakes and Dairy	73.73 kB	1.2 K	225.00 B	1	49.15 kB
Beauty and Hygiene	2.58 MB	27 K	235.00 B	1	442.37 kB
Beverages	65.54 kB	1 K	216.00 B	1	49.15 kB
Cleaning and Household	282.62 kB	3.9 K	240.00 B	1	94.21 kB

Figura 19 - Base de dados Big Basket no MongoDB

Document ID	Product Name	Brand	Price	Discount Price	Quantity	Category	SubCategory
ObjectId('6370332a80184c0259db682ef')	"Diaper Pants - XG"	"Pampers"	399	361	"16 pcs"	"Baby Care"	"Diapers"
ObjectId('6370332a80184c0259db682f0')	"All-Round Protection Diaper Pants - XXL, 15-25 kg, Ultra Absorb Core, ..."	"Pampers"	1199	841.5	"42 pcs"	"Baby Care"	"Diapers"
ObjectId('6370332a80184c0259db682f1')	"Premium Care Diapers - Large"	"Pampers"	2299	1559.5	"88 pcs"	"Baby Care"	"Diapers"

Figura 20 - Dados armazenados da categoria Baby Care no MongoDB

Nas três imagens acima temos a ligação da base de dados do *MongoDB* (figura 18) que permite que a base de dados esteja ativa e que o *output* do *PDI* tenha onde armazenar os dados, a segunda imagem (figura 19) tem aberto a base de dados *Big Basket* e também as categorias lá inseridas, tais como a *Baby Care* e a *Beverages...*, e por fim a última imagem (figura 20) tem aberto a categoria *Baby Care* e apresenta os primeiros 3 artigos e respetivos atributos desta categoria.



### *TransformationCheapestByCategory*

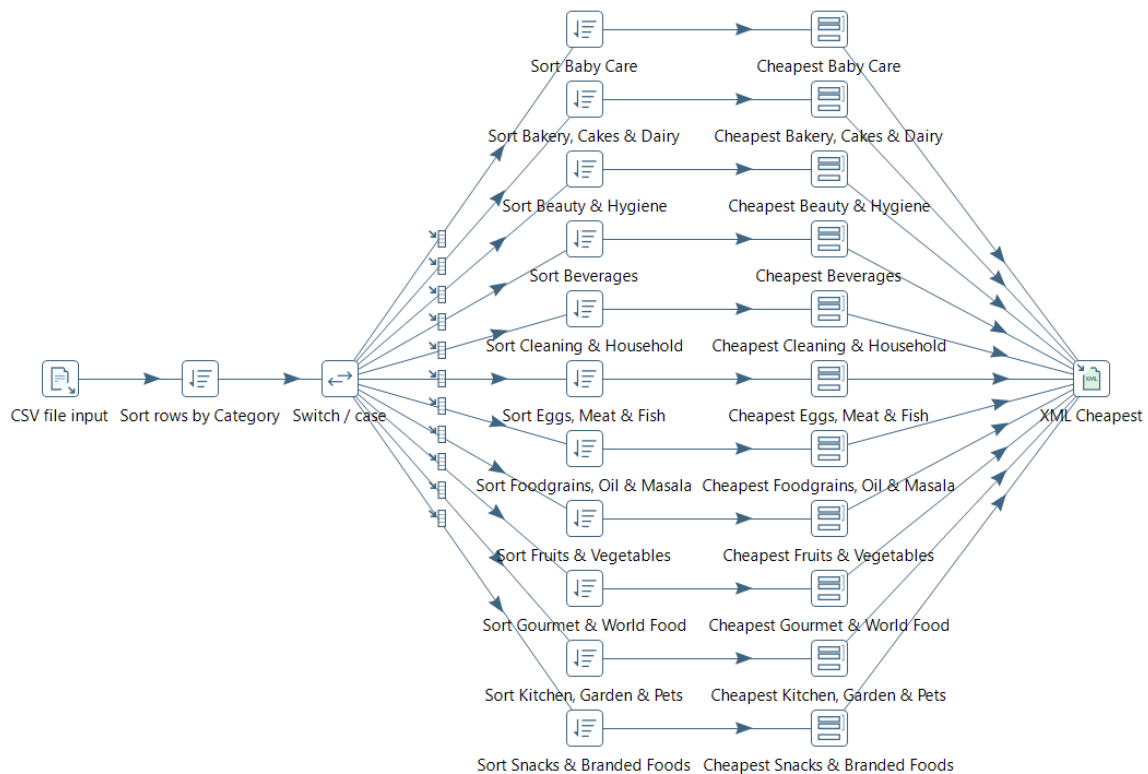


Figura 21 – TransformationCheapestByCategory

Nesta transformação, até ao passo do *Sort* após o *Switch/ case* faz todas as operações da transformação explicada no ponto acima porem, o *Sort* faz uma filtração para obter o artigo mais barato por categoria, depois exportando para um *.xml*.

Sort rows

Nome do Step
Sort Baby Care

Sort directory
%%java.io.tmpc

TMP-file prefix
out

Sort size (rows in memory)
1000000

Free memory threshold (in %)

Compress TMP Files?
☐

Only pass unique rows? (verifies keys only)
☒

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort
1	Price	S	N	N

Figura 22 - Sort Baby Care por preço ascendente



Figura 23 - Sample Row Baby Care mais barato

Sample rows

Step name: Cheapest Baby Care

Lines range: 1

Line nr fieldname:

Buttons: Help, OK, Cancela

Figura 24 - Sample rows do primeiro selecionado

Neste *Sample rows* é selecionado o que está inserido na primeira linha da coluna do *Sort Row* anterior, sendo este o artigo mais barato.



Figura 25 - XML Cheapest

Este *XML output*, tal como os anteriores, guarda um ficheiro *.xml* na pasta de Ficheiros de Data.

### **TransformationMostExpensiveByCategory**

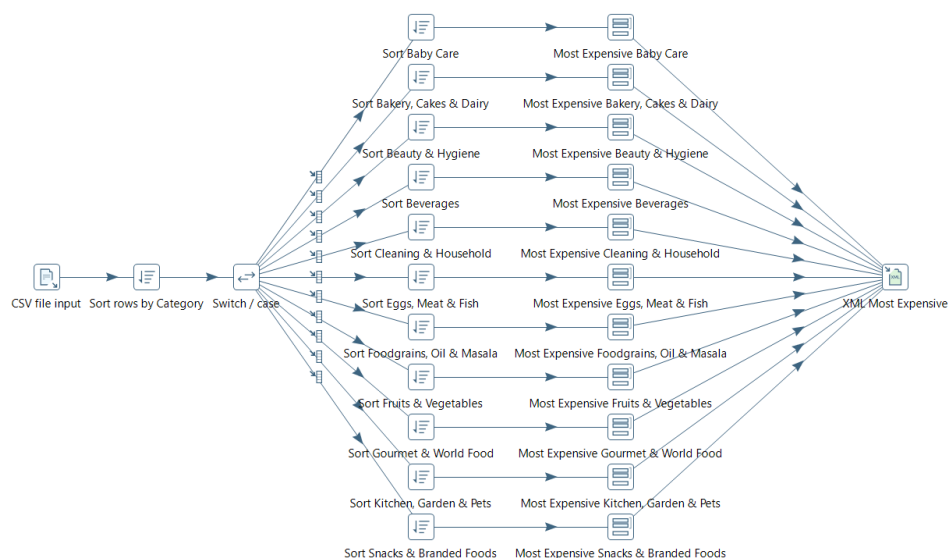


Figura 26 – TransformationMostExpensiveByCategory

Esta transformação faz exatamente o que a anterior faz, mas em vez do Sort estar por ordem ascendente, esta por ordem decrescente, ficam com a primeira linha com o artigo mais caro.

## 4.2. Jobs

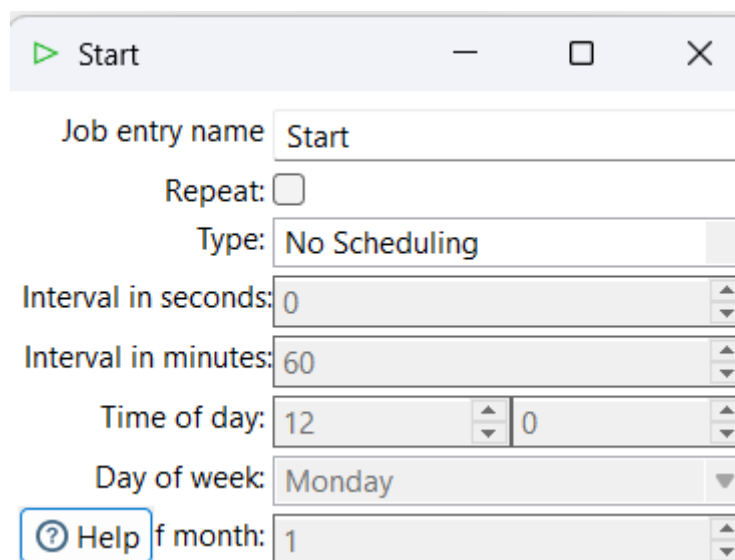
O *Job* é o que orquestra as atividades *ETL* do *PDI*.

Este projeto tem dois *Jobs*. O primeiro que cria páginas HTML com os dados das tabelas e o segundo que envia um email com os *logs*.

### *Job Send Email*



Figura 27 - Job Send Email



The screenshot shows a window titled 'Start' with a green play button icon. The configuration fields are as follows:

- Job entry name: Start
- Repeat: ☐
- Type: No Scheduling
- Interval in seconds: 0
- Interval in minutes: 60
- Time of day: 12 0
- Day of week: Monday
- Help button (circled in blue) and f month: 1

Figura 28 – Start

Com isto inicia o *Job*.

Transformation

Entry Name:  
Transformation

---

Transformation:  
C:/Users/ossie/Desktop/ISI/Trabalho/TransformationSortItemCategory.ktr Browse...

Figura 29 - Transformação do SortItemCategory

Isto corre a transformação que separa os artigos por categoria.

Mail validator

Job entry name Mail validator

Email address a21116@alunos.ipca.pt

Settings

SMTP check? ☐

Time out 0

Email sender noreply@domain.com

Default SMTP

Help OK Cancela

Figura 30 - Mail validator

Este passo valida o email utilizado no processo seguinte.

Mail

Name of mail job Mail

Addresses Server EMail Message Attached Files

Destination

Destination a21116@alunos.ipca.pt

Cc:

BCc:

Sender

Sender name Eu Proprio

Sender address: a21116@alunos.ipca.pt

Help OK Cancela

Figura 31 - Email de envio e destino

Email de envio e destino.

Figura 32 - Autenticação do SMTP Server e email em uso

Dados do SMTP Server do Outlook e autenticação do meu email.

```

Job:
-----
JobName   : Job Send Email
Directory : /
JobEntry  : Mail

Previous results:
-----
Job entry Nr      : 1
Errors           : 0
Lines read       : 0
Lines written    : 0
Lines input      : 0
Lines output     : 0
Lines updated    : 0
Lines rejected   : 0
Script exist status : 0
Result          : true

Path to this job entry:
-----
Job Send Email
Job Send Email : : start : Start of job execution (2022/11/12 18:19:28.582)
Job Send Email : : Start : start : Start of job execution (2022/11/12 18:19:28.583)
Job Send Email : : Start : [nr=0, errors=0, exit_status=0, result=true] : Job execution finished (2022/11/12 18:19:28.584)
Job Send Email : : Transformation : Followed unconditional link : Start of job execution (2022/11/12 18:19:28.585)
  
```

Figura 33 - Email criado pelo Job

## Job HTML

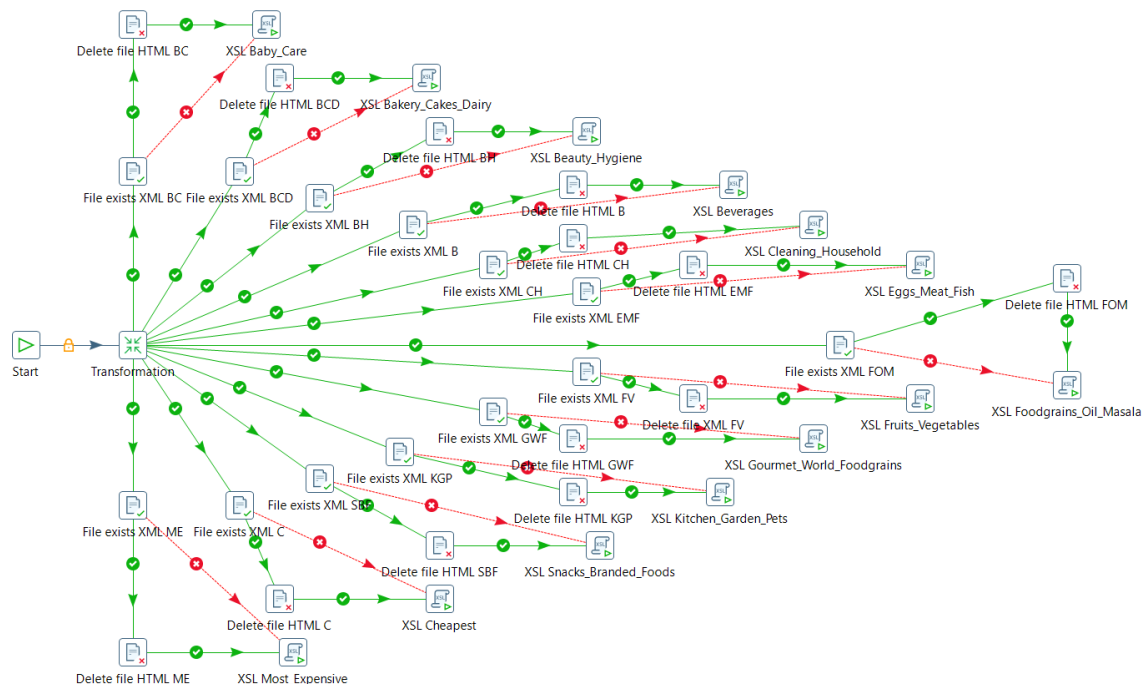


Figura 34 - Job HTML

Neste Job é utilizado a transformação “SortItemCategory” e a partir dos ficheiros .xml é criado páginas HTML.

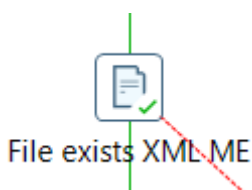


Figura 35 - File exists XML ME (Most Expensive)

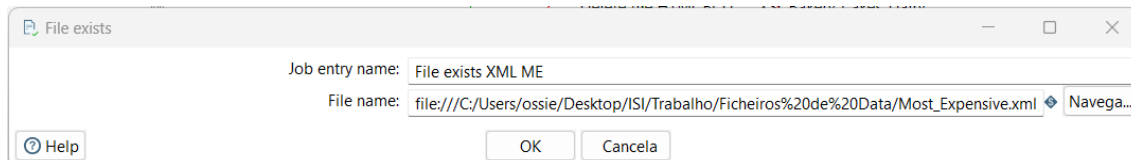


Figura 36 - File exists XML ME função

No File Exists o Job procura se o ficheiro .xml existe. Caso ele exista ele apaga o ficheiro HTML. Se não existir cria o ficheiro HTML.



### Delete file HTML ME

Figura 37 - Delete file HTML ME

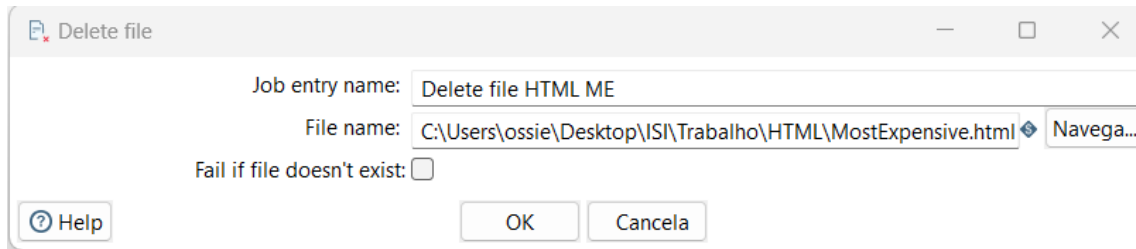


Figura 38 - Delete file HTML ME função

Aqui o ficheiro *.html* é eliminado.



### XSL Most\_Expensive

Figura 39 - XSL Most\_Expensive

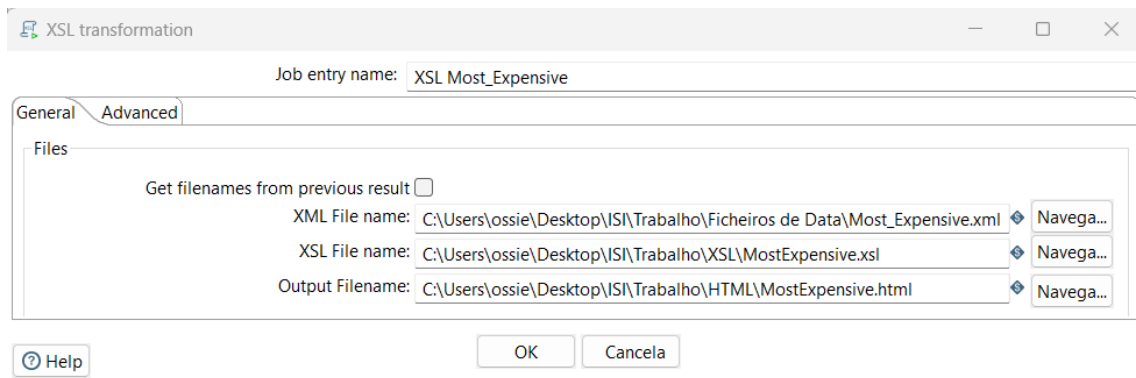


Figura 40 - XSL Most\_Expensive função

Caso o ficheiro de HTML não existe, esta função procura o ficheiro *.xml* e *.xsl* e com esses dados consegue criar o ficheiro *.html*.

BabyCare.html

Ficheiro | C:/Users/ossie/Desktop/ISI/Trabalho/HTML/BabyCare.html

Email IPCA Moodle GitHub Azure DevOps Scan Arte Android | what3wor... what3words/w3w-a... 21 Best Cheap Euro... 10 Best Winter Vaca...

**Baby Care**

Product Name	Brand	Price	Discount Price	Quantity	Category	SubCategory
Diaper Pants - XXL	Pampers	399.00	361.00	16 pcs	Baby Care	Diapers
All-Round Protection Diaper Pants - XXL, 15-25 kg, Ultra Absorb Core, Leakage Prevention for upto 12 Hours	Pampers	1199.00	841.50	42 pcs	Baby Care	Diapers
Premium Care Diapers - Large	Pampers	2299.00	1559.50	88 pcs	Baby Care	Diapers
Premium Care Diapers - Large	Pampers	320.00	290.50	13 pcs	Baby Care	Diapers
Premium Care Diaper Pants - Large, 9-14 kg, Air Channels, Lotion with Aloe Vera	Pampers	799.00	610.00	30 pcs	Baby Care	Diapers
Premium Care Diaper Pants - Large, 9-14 kg, Lotion with Aloe Vera	Pampers	3597.00	2311.00	132 pcs	Baby Care	Diapers
Premium Care Diapers - Large	Pampers	1199.00	862.00	44 pcs	Baby Care	Diapers
Premium Care Diapers - Extra Large	Pampers	3597.00	2345.50	108 pcs	Baby Care	Diapers
Premium Care Diaper Pants - XL, 12-17 kg, Lotion with Aloe Vera	Pampers	2299.00	1612.00	72 pcs	Baby Care	Diapers
Premium Care Diaper Pants - Extra Large, 12-17 kg, Lotion with Aloe Vera	Pampers	299.00	258.00	11 pcs	Baby Care	Diapers
Premium Care Diaper Pants - Extra Large, 12-17 kg, Lotion with Aloe Vera	Pampers	799.00	626.50	24 pcs	Baby Care	Diapers
Premium Care Diaper Pants - XL, 12-17 kg, Lotion with Aloe Vera	Pampers	1199.00	867.50	36 pcs	Baby Care	Diapers
Diaper Pants - Medium	Pampers	1199.00	821.00	76 pcs	Baby Care	Diapers
All-Round Protection Diaper Pants - M, 7-12 kg, Anti-Rash, Ultra Absorb	Pampers	3149.00	1858.00	200 pcs	Baby Care	Diapers
All-Round Protection Diaper Pants - M, 7-12 kg, Anti-Rash, Ultra Absorb, Leakage Prevention for upto 12 Hours	Pampers	2399.00	1455.50	152 pcs	Baby Care	Diapers
Diaper Pants - Medium	Pampers	399.00	361.00	26 pcs	Baby Care	Diapers
All-Round Protection Diaper Pants - Medium, 7-12 kg, Anti-Rash, Ultra Absorb	Pampers	799.00	609.50	50 pcs	Baby Care	Diapers
Baby Lotion	Himalaya Baby	170.00	122.00	200 ml	Baby Care	Baby Creams & Lotions
Baby Lotion	Himalaya Baby	300.00	219.50	400 ml	Baby Care	Baby Creams & Lotions
Baby Lotion	Himalaya Baby	95.00	90.25	100 ml	Baby Care	Baby Creams & Lotions
Premium Care Diaper Pants - Medium, 7-12 kg, Air Channels, Lotion with Aloe Vera	Pampers	320.00	290.00	16 pcs	Baby Care	Diapers
Premium Care Diaper Pants - M, 7-12 kg, Cotton-Like Softness, Lotion with Aloe Vera	Pampers	1199.00	908.00	54 pcs	Baby Care	Diapers
Premium Care Diaper Pants - M, 7-12 kg, Lotion with Aloe Vera	Pampers	2398.00	1567.00	108 pcs	Baby Care	Diapers
Premium Care Diaper Pants - Medium, 7-12 kg, Air Channels, Lotion with Aloe Vera	Pampers	799.00	639.00	38 pcs	Baby Care	Diapers
Premium Care Diaper Pants - Medium, 7-12 kg, Air Channels, Lotion with Aloe Vera	Pampers	3449.00	2276.50	162 pcs	Baby Care	Diapers
Baby Powder	Himalaya Baby	235.00	171.00	400 g	Baby Care	Baby Powder

Figura 41 - HTML gerado pelo Job



## 5. Conclusão

A integração de dados forneceu uma solução prática para um problema complexo.

Atualmente, conseguimos utilizar uma aplicação, e efetuar o envio de dados para outra aplicação completamente diferente, mediante um conjunto de padrões associados. No entanto, e dado o elevado risco de segurança, trabalhar com dados online é um trabalho bastante difícil e complexo, uma vez que potencia a exploração de falhas de segurança, isto é injeção de código.... Nesse seguimento, as ferramentas ETL tornam-se uma necessidade, não só pela redução no tempo despendido a criar a solução, mas também a extrair informação e transformar a mesma.

## 6. Referências

Como referências foi utilizado o material fornecido pelo professor e também:

<https://www.kaggle.com/datasets/chinmayshanbhag/big-basket-products>