

2023-2024

P.PORTO

**ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO**

PROJETO DE PROCESSAMENTO ESTRUTURADO DE INFORMAÇÃO

TRABALHO PRÁTICO – ÉPOCA NORMAL

Trabalho elaborado por:

Grupo 19

8220169 – César Ricardo Barbosa Castelo

8220337 – Hugo Ricardo Almeida Guimarães

8220307 – Pedro Marcelo Santos Pinho

Índice

Índice de Figuras	2
Chave de Siglas	3
Introdução	4
Abordagem	5
1. Mongo DB	6
a. Estrutura da base de dados	6
a. Clientes	6
b. Produtos	7
c. Vendas	7
d. Devoluções	8
b. Importação dos dados fornecidos	9
c. Integração do BaseX com o MongoDB	10
2. Organização do XML	12
a. Regras (XML Schema)	12
MongoDB Charts	13
Apreciação Crítica	14

Índice de Figuras

Figura 1 - coleções do mongodb.....	6
Figura 2 - Estrutura da coleção cliente no mongoDB.....	6
Figura 3 - Estrutura da coleção produtos no mongoDB.....	7
Figura 4 - estrutura da coleção vendas no mongodb	7
Figura 5 - Estrutura da coleção devoluções no mongodb	8
Figura 6 - script inicial para criação da base de dados e coleções.....	9
Figura 7 - script para a criação de índices	9
Figura 8 - pipelines para a transformação dos dados	9
Figura 9 - Relatório de vendas devolvido pelo BaseX	10
Figura 10 - Relatório de devoluções devolvido pelo baseX.....	11
Figura 11 - Ficheiros com as regras para montar as regras dos relatórios.....	12
Figura 12 - Regras dos relatórios	12
Figura 13 - Gráficos criados no MongoDB Charts	13

Chave de Siglas

XML	Extensible Markup Language
XSD	XML Schema Definition
CSV	Comma Separated Values
JSON	Java Script Object Notation

Introdução

Este trabalho foi realizado para o âmbito da disciplina de *Processamento Estruturado de Informação*, que funcionará como integrador dos conhecimentos adquiridos no decorrer das aulas.

Este trabalho consiste na construção de um vocabulário XML para suportar o envio de relatórios de vendas e devoluções de cada parceiro da empresa *Phone for You*, uma empresa que comercializa smartphones através de várias lojas. Assim cada parceiro terá de implementar nos seus sistemas informáticos, um módulo que suporta a geração de documentos XML de vendas e devoluções de acordo com vocabulário estabelecido.

Para além da construção de um vocabulário XML para suportar o envio de relatórios de vendas e devoluções, também será preciso fazer uma transformação dos dados já existentes de vários ficheiros CSV para uma base de dados orientada a documentos (MongoDB), como também será necessário fazer uma API para aceder a esses mesmos dados, e construir os relatórios para depois retornar ao utilizador.

Abordagem

A elaboração deste relatório seguiu uma abordagem metodológica cuidadosa e estruturada. Onde inicialmente, foram identificadas as necessidades específicas do projeto, como a elaboração de regras, a criação de uma base de dados com todas as informações dadas, e a construção de uma API para poder ter acesso aos dados da base de dados, tentando delinear os requisitos e objetivos fundamentais a serem alcançados, visando ao máximo com que todos os elementos do grupo tivessem a mesma visão sobre o projeto, para que se pudesse trabalhar de forma eficaz, e possibilitando a entrega de um trabalho homogéneo.

1. Mongo DB

a. Estrutura da base de dados

Para a realização deste trabalho, foi-nos enviado vários ficheiros, em formato CSV, que representam um subconjunto de informação tipicamente armazenada por um dos parceiros da *Phone for You*, com o objetivo de fazer a importação deles para o Mongo DB. No entanto, como a informação de ficheiros estava muito fragmentada, uma importação direta era inviável, já que as consultas iriam demorar imenso tempo, então para resolver esse problema, fez-se uma transformação dos dados quando estes foram importados. Os dados ficaram divididos segundo as seguintes coleções:

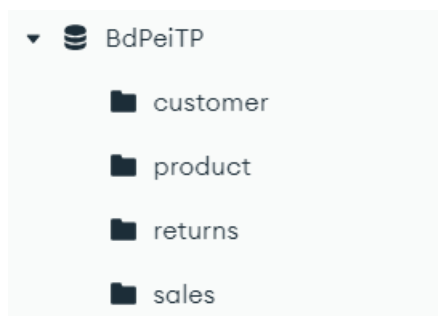


FIGURA 1 - COLEÇÕES DO MONGODB

a. Clientes

```
_id: ObjectId('65819ebb6eb35e1f6b976b0d')
first_name: "MARY"
last_name: "SMITH"
email: "MARY.SMITH@sakilacustomer.org"
ative: 1
create_date: "2021-02-14 22:04:36.000"
gender: "M"
birthDate: 1969-01-29T00:00:00.000+00:00
address_info: Object
  address: "1913 Hanoi Way"
  address2: null
  district: " "
  postal_code: 35200
  city: "Sasebo"
  country: "Japan"
customer_id: 1
```

FIGURA 2 - ESTRUTURA DA COLEÇÃO CLIENTE NO MONGODB

Para poder importar informação dos clientes, foi preciso juntar vários ficheiros em uma única coleção, sendo esses ficheiros: “address.csv”, “city.csv” e “country.csv”.

Como um cliente tem uma morada registada, e cada morada tem de ter registado uma cidade, e uma cidade está registada a um país, decidiu-se juntar a informação dessas tabelas em uma única coleção, de forma que não fosse preciso percorrer muitas coleções apenas para obter informações de morada.

b. Produtos

```

_id: ObjectId('65819f8b6eb35e1f6b977508')
product_id: 963
list_price: 24762
brand: "xiaomi"
model: "Xiaomi Redmi Note 12 Pro 5G"
5g: 1
processor_brand: "dimensity"
battery_capacity: 5000
fast_charging: 1
ram_capacity: 6
internal_memory: 128
screen_size: 6.67
os: "android"
primary_camera: 50
categories: Array (5)
  0: Object
    name: "Camera Quality"
    sub_categories: Array (1)
      0: "Good Cameras"
  1: Object
    name: "Storage Capacity"
    sub_categories: Array (1)
      0: "High Storage"

```

FIGURA 3 - ESTRUTURA DA COLEÇÃO PRODUTOS NO MONGODB

Para os produtos aconteceu algo semelhante à coleção dos clientes, no entanto em vez de se juntar a morada, juntou-se o produto com as categorias, visto que um produto tem uma ou mais categorias, e uma categoria pode ter uma ou mais subcategorias. Juntando assim a informação dos ficheiros “Product.csv”, “sub_category_product.csv”, “sub_category.csv” e “category.csv”.

c. Vendas

```

_id: ObjectId('659c384703f7b3dcf9668a4e')
invoice_id: 1000019
date: 2022-02-04T00:00:00.000+00:00
customer: Object
  _id: ObjectId('659c038503f7b3dcf96686bd')
  first_name: "TIMOTHY"
  last_name: "BUNN"
  address_info: Object
    customer_id: 325
sales_lines: Array (5)
  0: Object
    _id: ObjectId('659c39ab03f7b3dcf96b03a9')
    id: 39230
    total_with_vat: 1639.3
    quantity: 1
    product_id: 96
    invoice_id: 1000019
  1: Object
  2: Object
  3: Object
  4: Object
totalSales: 6
totalRevenue: 3418.0633
products_on_sales: Array (5)
  0: Object
    _id: ObjectId('659bcdde03f7b3dcf9599390')
    product_id: 96
    brand: "huawei"
    categories: Array (5)

```

FIGURA 4 - ESTRUTURA DA COLEÇÃO VENDAS NO MONGODB

Para poder criar a coleção das vendas, foi preciso unir os ficheiros: “sales_header.csv”, “sales_lines.csv”, pois, uma “sales_header” pode ter uma ou mais “sales_lines”. Para além de unir esses dois ficheiros, também se colocou alguma informação do cliente que comprou o produto, e dos produtos que o mesmo comprou, assim evita-se ter de percorrer várias

coleções para quando se for criar relatórios, fazendo com a aplicação tenha um desempenho muito melhor.

d. Devoluções

```
_id: ObjectId('659bcde903f7b3dcf9599706')
invoice_id: 1000021
product_id: 664
date: 2022-05-27T00:00:00.000+00:00
▼ product: Object
  _id: ObjectId('659bcde03f7b3dcf95995c8')
  brand: "samsung"
  model: "Samsung Galaxy S21 FE 5G"
  ▼ categories: Array (5)
    ▼ 0: Object
      name: "Performance"
      ▼ sub_categories: Array (1)
        ▼ 1: Object
        ▼ 2: Object
        ▼ 3: Object
        ▼ 4: Object
    ▼ sale: Object
      invoice_id: 1000021
      date: 2022-05-27T00:00:00.000+00:00
      daysUntilReturn: 0
      earlyReturn: true
  ▼ customer: Object
    _id: ObjectId('659c038503f7b3dcf966864d')
    customer_id: 213
    first_name: "GINA"
    last_name: "WILLIAMSON"
    ▼ address_info: Object
```

FIGURA 5 - ESTRUTURA DA COLEÇÃO DEVOLUÇÕES NO MONGODB

Nas devoluções, juntou-se um pouco de toda a informação das outras coleções criadas anteriormente, pois, uma devolução é efetuada por um cliente, sobre um produto em específico, e esse produto está associado a uma venda a partir do código da sua fatura (invoice_id).

Uma característica do MongoDB, em comparação com outros modelos de bases de dados, é a sua abordagem semiestruturada. Esta abordagem confere flexibilidade ao MongoDB, que se aproveita da redundância para proporcionar um alto desempenho. Por exemplo, na coleção de clientes, cada cliente possui informações de endereço armazenadas em "address_info". Notavelmente, o mesmo cliente também terá informações à cerca do endereço registadas nas devoluções. Caso o cliente mude de endereço, a alteração será efetuada apenas na coleção clientes, sem afetar as informações nas devoluções. Este design flexível e eficiente destaca a capacidade do MongoDB de lidar com dados semiestruturados de maneira ágil e eficaz, por isso tentou-se ao máximo durante as migrações dos dados abusar da redundância, para que as consultas conseguissem ser feitas de forma quase instantânea.

b. Importação dos dados fornecidos

Para importar os dados fornecidos para a base de dados, primeiramente, foi preciso criar um script que criasse uma base de dados vazia no MongoDB, e depois criasse várias coleções, que irão conter os dados dos ficheiros csv, esse ficheiro chama-se “*Configuracao Inicial da base de dados.txt*”, e ele está na pasta “.*\MongoDB*”


 Configuracao inicial da base de dados.txt

FIGURA 6 - SCRIPT INICIAL PARA CRIAÇÃO DA BASE DE DADOS E COLEÇÕES

Nota: o ficheiro apenas cria coleções vazias, para fazer a importação dos dados dos ficheiros é preciso fazê-lo manualmente a partir do MongoDB Atlas.

Após ter todas as coleções com os dados originais importados, vai ser preciso executar o script de outro ficheiro chamado “*Indices depois do import do csv.txt*”, ele também está localizado na pasta “.*\MongoDB*”, e tal como o nome indica, ele vai definir os índices necessários para as migrações que se vai fazer, em alguns casos, as migrações nem são possíveis de realizar sem estes índices


 Indices depois do import do csv.txt

FIGURA 7 - SCRIPT PARA A CRIAÇÃO DE ÍNDICES

Com os dados importados e os índices criados, vai ser preciso fazer uma migração desses dados para novas coleções, que conterão os dados organizados de maneira que o MongoDB consiga ler de forma eficiente. Para isso, dentro da pasta “.*\MongoDB\pipelines*” contém cinco scripts, onde quatro deles contêm as pipelines de agregação¹ necessárias para a migração dos dados, o último são alguns scripts para adicionar alguns campos adicionais em algumas coleções. Os ficheiros estão numerados de 1 a 5 para indicar que eles devem ser executados por ordem crescente, dessa forma, permite que os últimos ficheiros, que contêm mais informação e necessitam de mais alterações, não precisem de fazer tantas transformações, basta aceder às coleções que já têm os dados prontos para poder fazer a transferência, poupando assim bastante tempo e esforço.






 1. Customer_csv para Customer.txt 2. Product_csv para Product.txt 3. sales_header e lines para Sales.txt 4. returns_csv para returns.txt 5. campos adicionais.txt

FIGURA 8 - PIPELINES PARA A TRANSFORMAÇÃO DOS DADOS

¹ Um pipeline de agregação consiste em um ou mais estágios que processam documentos, onde cada estágio executa uma operação nos documentos de entrada

c. Integração do BaseX com o MongoDB

Para a integração do BaseX com o MongoDB, foram desenvolvidas duas consultas na data API do MongoDB, para receber relatórios de vendas e devoluções para um determinado mês. Para poder aceder a essas consultas, foi desenvolvida uma API no BaseX que comunica com a data API do MongoDB, que devolve os dados e transforma-os para o formato xml. É graças a esta API que é possível especificar o mês e o ano dos dados que queremos receber, pois é nela que eles são definidos como parâmetros.

The screenshot displays the BaseX web interface for a REST client. At the top, the breadcrumb is 'Trabalho PEI / sale report baseX'. The request method is 'GET' and the URL is 'http://localhost:8080/sales?ano=2022&mes=12'. The 'Query Params' table shows 'ano' as 2022 and 'mes' as 12. The status bar indicates 'Status: 200 OK', 'Time: 4.06 s', and 'Size: 347.97 KB'. The response is shown in 'Pretty' XML format.

Key	Value	Description	Bulk Edit
ano	2022		
mes	12		

```

1 <sales>
2   <sale>
3     <invoice_id>1001156</invoice_id>
4     <date>2022-12-13T00:00:00Z</date>
5     <sales_lines>
6       <sale_line>
7         <id type="number">40417</id>
8         <total_with_vat>137.9885</total_with_vat>
9         <quantity type="number">1</quantity>
10        <product_id>570</product_id>
11      </sale_line>
12    </sales_lines>
13  </sale>
14  <sale>
15    <invoice_id>100132</invoice_id>
16    <date>2022-12-23T00:00:00Z</date>
17    <sales_lines>
18      <sale_line>
19        <id type="number">249459</id>
20        <total_with_vat>146.761</total_with_vat>
21        <quantity type="number">1</quantity>
22        <product_id>978</product_id>
  
```

FIGURA 9 - RELATÓRIO DE VENDAS DEVOLVIDO PELO BASEX

Trabalho PEI / returns Report baseX

GET http://localhost:8080/returns?ano=2022&mes=12

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> ano	2022			
<input checked="" type="checkbox"/> mes	12			

Body Cookies Headers (5) Test Results Status: 200 OK Time: 2.17 s Size: 1 MB Save as example

Pretty Raw Preview Visualize XML

```
1 <returns>
2   <return>
3     <invoice_id>1002707</invoice_id>
4     <product_id>945</product_id>
5     <daysUntilReturn type="number">7</daysUntilReturn>
6     <earlyReturn type="boolean">>false</earlyReturn>
7     <date>2022-12-11T00:00:00Z</date>
8     <sale>
9       <invoice_id>1002707</invoice_id>
10      <date>2022-12-04T00:00:00Z</date>
11    </sale>
12    <customer>
13      <customer_id>288</customer_id>
14      <first_name>BOBBIE</first_name>
15      <last_name>CRAIG</last_name>
16      <address_info>
17        <country>Hong Kong</country>
18        <city>Kowloon and New Kowloon</city>
19        <address_info>58124</address_info>
20      </address_info>
21    </customer>
22    <product>
```

FIGURA 10 - RELATÓRIO DE DEVOLUÇÕES DEVOLVIDO PELO BASEX

2. Organização do XML

a. Regras (XML Schema)

Para as regras do XML, tentou-se ao máximo com que elas fossem o mais reutilizáveis possíveis, então criou-se vários ficheiros, onde cada um deles define regras específicas para um componente, como pode ser visto na Figura 11:

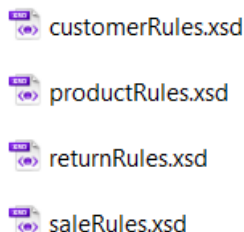


FIGURA 11 - FICHEIROS COM AS REGRAS PARA MONTAR AS REGRAS DOS RELATÓRIOS

Estas regras encontram-se na pasta “.\BaseX\xsd\rules”.

Por estarem em ficheiros diferentes, possibilita com que seja possível montar um relatório customizado, como por exemplo: um relatório de vendas; devoluções; vendas e devoluções; etc...

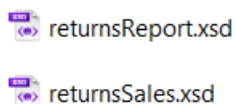


FIGURA 12 - REGRAS DOS RELATÓRIOS

Os ficheiros mostrados na figura 12 são os ficheiros com as regras usadas na API do BaseX para validar os resultados obtidos. Estes ficheiros encontram-se na pasta “.\BaseX\xsd”.

MongoDB Charts

Como o MongoDB fornece uma interface intuitiva e personalizável, permitindo a criação rápida de gráficos interativos, decidiu-se usá-la neste projeto para criar gráficos a partir dos valores lá armazenados. Desta forma podemos ter um guia visual, de como uma informação se relaciona com a outras, permitindo uma maior análise sobre o negócio, esta é uma estratégia muito usada por lojas e empresas para identificarem e superarem adversidades. Em última análise, a escolha do MongoDB Charts representa uma abordagem eficaz para maximizar a utilidade dos dados disponíveis e impulsionar o sucesso de um projeto



FIGURA 13 - GRÁFICOS CRIADOS NO MONGODB CHARTS

Link para poder aceder aos gráficos: <https://charts.mongodb.com/charts-project-0-voykk/public/dashboards/659fddb5-26de-44b5-821b-79606e9900af>

Apreciação Crítica

Ao longo deste relatório, foi mostrado como foi realizada a migração dos dados em texto para uma base de dados orientada a ficheiros, como foram feitas as consultas, e como elas estão organizadas de forma a ficarem de acordo com o vocabulário XML desenvolvido.

Ao longo da realização do projeto deparamo-nos com vários desafios, como por exemplo, a migração dos dados dos ficheiros CSV para o MongoDB, a manipulação de datas e horas, e a passagem dos relatórios de JSON para XML a partir do BaseX, que de acordo com todos os elementos do grupo, foi o que deu mais trabalho. Infelizmente, alguns dos desafios encontrados não conseguiram ser ultrapassados, como: a identificação do parceiro que gerou o relatório de vendas e devoluções, e a inserção de alguns campos adicionais.

Mas, mesmo com todos esses desafios, podemos concluir que a realização deste trabalho foi bem concebida tendo em conta o que nos foi proposto e também as nossas próprias exigências para o trabalho, entre elas, a constante busca pela estrutura da base de dados, que conseguisse gerar relatórios da forma mais eficiente possível. Este projeto mostrou-se muito importante para o nosso desenvolvimento, pois conseguimos colocar em prática aquilo que foi lecionado durante as aulas, assim, sedimentando os nossos conhecimentos.