Relatório do Trabalho de Processamento Estruturado de Informação

Bruno Miguel Ferreira – Nº 8220191

Miguel Rangel Tavares – Nº 8220229

Grupo 11

Escola Superior de Tecnologia e Gestão

Instituto Politécnico do Porto

Data: 19 de Janeiro de 2024

Índice

Introdução

No contexto da disciplina de Processamento Estruturado de Informação (PEI), este projeto emerge como resposta às necessidades da "Phone for You", uma empresa que comercializa smartphones através de várias lojas, contando com diversos parceiros nacionais para a venda de seus produtos. A crescente expansão da empresa motivou a demanda por relatórios mensais detalhados de vendas e devoluções por parte de cada parceiro.

Com o intuito de facilitar essa integração de dados, optou-se por disponibilizar um vocabulário XML que servirá como padrão para a geração desses relatórios. Este vocabulário abrangerá informações cruciais sobre vendas e devoluções, proporcionando uma visão abrangente do desempenho de cada parceiro.

Particularmente, este projeto concentra-se na implementação de módulos nos sistemas informáticos de cada parceiro para exportar informações detalhadas sobre vendas e devoluções, utilizando requisições HTTP e APIs como base para a comunicação eficiente. A estruturação destes dados será realizada em conformidade com o vocabulário XML definido.

Ao explorar a dinâmica das requisições HTTP e APIs, buscamos não apenas atender aos requisitos específicos apresentados no enunciado, mas também desenvolver uma solução robusta que possa ser aplicada em cenários reais. Este trabalho engloba não só a manipulação e exportação de dados, mas também a estruturação eficaz desses dados em um banco de dados MongoDB, seguida pela geração de relatórios XML utilizando a ferramenta BaseX.

No decorrer deste relatório, detalharemos cada etapa do processo de desenvolvimento, desde a importação dos dados fornecidos em formato CSV para o MongoDB até à implementação da API em BaseX para a exportação de relatórios em XML. Espera-se que este projeto demonstre a aplicação prática de conceitos fundamentais aprendidos na disciplina de PEI.

Ferramentas utilizadas

Como mencionado anteriormente, para a realização deste projeto focamo-nos na utilização de 3 ferramentas principais: Postman, MongoDB e BaseX.

Postman

Uma plataforma de colaboração para o desenvolvimento e teste de APIs Application programming interfaces). É uma ferramenta que oferece um ambiente amigável e intuitivo para criar, testar e documentar APIs de maneira eficiente. Algumas das principais funcionalidades do Postman incluem: Criação de Requisições HTTP que permite criar diferentes tipos de solicitações HTTP, como GET, POST, PUT, DELETE, entre outras, facilitando a interação com APIs; Ambientes e Variáveis que permite a criação de ambientes para diferentes cenários, possibilitando a configuração de variáveis que podem ser usadas em várias solicitações; Colaboração que oferece recursos de colaboração, permitindo que equipas trabalhem em conjunto.

MongoDB

O MongoDB é uma base de dados direcionada a documentos que se destaca pela sua flexibilidade e escalabilidade. Ele armazena dados em documentos no formato BSON (Binary JSON) e não requer um esquema fixo, proporcionando uma abordagem dinâmica para a modelagem de dados.

BaseX

O BaseX é um sistema de gerenciamento de banco de dados XML (Extensible Markup Language) que dá prioridade à eficiência e a manipulação otimizada de dados no formato XML. Especializado no armazenamento e gerenciamento de informações em documentos XML, o BaseX destaca-se pelas suas capacidades avançadas. Utilizando a linguagem de consulta XQuery, oferece suporte a consultas declarativas, permitindo a extração eficiente de dados complexos.

Utilização do MongoDB

Numa fase inicial do projeto, o grupo tratou de gerir os dados e organizá-los de forma a facilitar futuras pesquisas através da API.

Após a criação de um cluster no Mongo Atlas, foi criada a database "Projeto" onde foram criadas diversas collections. Cada collection guardava os documentos de cada ficheiro ".csv" fornecido no enunciado.

De seguida foi posto em prática o conteúdo lecionado nas aulas ao utilizarmos a framework de agregação para embutir documentos, noutros documentos, de forma a facilitar a interpretação e a pesquisa dos dados.

Exemplo:

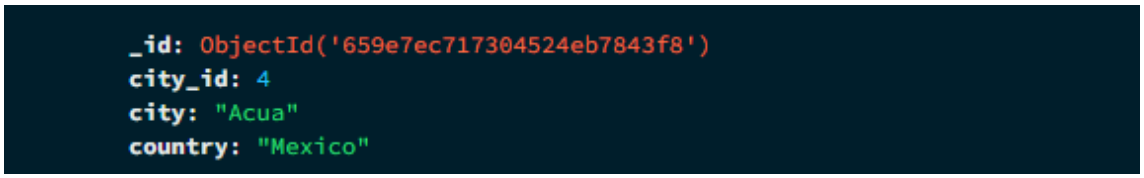

Figura 1 Exemplo de documento presente na collection city

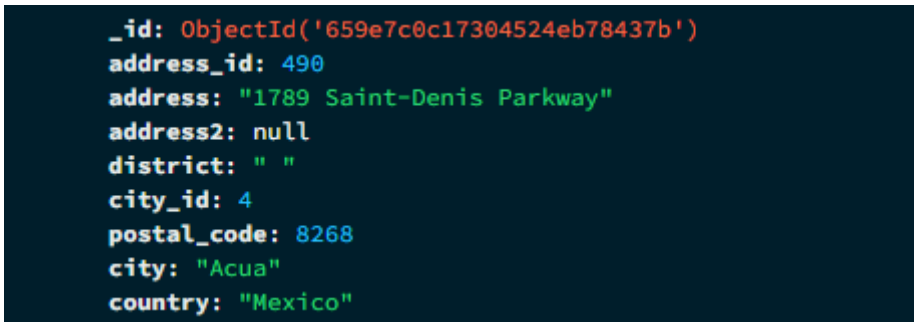

Figura 2 Exemplo do documento da Figura 1 embutido no documento da collection address que continha a respetiva referência

Os documentos foram organizados da seguinte forma:

Customer: Todos os documentos das collections "city" e "country" estão embutidos em "address", que por sua vez estão embutidos em "customer". Foi definido um campo "clientType" para responder aos requisitos do enunciado, nomeadamente, "o tipo de cliente com base no tempo que se encontra registado no sistema". E um campo "num_compras" para guardar o número de compras feitas pelo cliente nos últimos 3 anos.

Product: Todos os documentos da collection "product" têm embutido documentos "category" que por sua vez contêm documentos "sub_category_product".

Sales_header: Cada documento da collection "sales_header" armazena as respetivas: "sales_lines", "customer" e "products". Ainda foram adicionados os campos "total_sales", que armazena o valor total da compra, e "productsSold" que armazena os diferentes produtos daquela venda.

Foi criada uma sample “salesCollectionSample”.

Returns: Cada documento da collection “returns” armazena um array dos produtos devolvidos, e um objeto referente à venda devolvida. Foram adicionados os campos “early_return” e “days_to_return” para responder às seguintes questões colocadas no enunciado:

“Para cada devolução deverá ser apresentado o número de dias que passaram até à devolução do produto.
Indicador de devolução precoce, isto é, se o produto foi ou não devolvido no prazo de 3 dias após a compra.”

Todas as agregações utilizadas foram partilhadas nos ficheiros “.txt” enviados juntamente com o relatório.

Utilização do BaseX

De seguida, foram desenvolvidas as querys para retornar os dados solicitados no enunciado.

Através de ficheiros de consulta desenvolvidos que filtravam e devolviam a informação presente na base de dados do MongoDB, que previamente se encontrava em JSON, em XML.

Neste âmbito foram desenvolvidos dois ficheiros: o “salesQueryReport.xqm” e o “returnsQueryReport.xqm”.

Ficheiros XQM

Os seguintes ficheiros foram partilhados juntamente com o relatório.

Ficheiro “salesQueryReport”

Função getSales: Esta função é acessível através de uma chamada GET para o caminho "/getSales".

Aceita dois parâmetros de consulta: "ano" e "mes".

Internamente, chama getSalesRawData para obter dados brutos, transforma esses dados usando transformSalesXML e retorna o resultado.

Função getSalesRawData: Constrói uma URL, define parâmetros de consulta, cabeçalhos e corpo para enviar uma solicitação HTTP POST para uma API MongoDB.

Utiliza o módulo http para enviar a solicitação e retorna a resposta.

Funções de cálculo de valores:

getDistinctSalesCount: Retorna o número distinto de IDs de fatura nas vendas.

getTotalSalesValue: Retorna o valor total de vendas.

getDistinctProductCount: Retorna o número distinto de IDs de produtos nas vendas.

getDistinctCustomerCount: Retorna o número distinto de IDs de clientes nas vendas.

Função transformSalesXML: Transforma os dados brutos de vendas em um formato XML específico.

Inclui informações como NIF, Nome, Morada, Ano Fiscal, Mês, Número de Vendas Distintas, Valor Total de Vendas, Número Distinto de Clientes, e detalhes de cada venda.

O XML resultante possui uma estrutura hierárquica que inclui informações gerais e detalhes específicos de cada venda.

Ficheiro “returnsQueryReport”

Função getReturns: Esta função é acessível através de uma chamada GET para o caminho "/getReturns".

Aceita dois parâmetros de consulta: "ano" e "mes".

Internamente, chama getReturnsRawData para obter dados brutos, transforma esses dados usando returnsToXML e retorna o resultado.

Função getReturnsRawData: Constrói uma URL, define parâmetros de consulta, cabeçalhos e corpo para enviar uma solicitação HTTP POST para uma API MongoDB relacionada a devoluções.

Utiliza o módulo http para enviar a solicitação e retorna a resposta.

Função getDistinctProductCount: Retorna o número distinto de IDs de produtos nas devoluções.

Função returnsToXML: Transforma os dados brutos de devoluções num formato XML específico.

Inclui informações como NIF, Nome, Morada, Ano Fiscal, Mês, Número Distinto de Produtos, e detalhes de cada devolução.

Cada devolução inclui informações como ID da fatura, data de devolução, dias para devolução, se foi uma devolução antecipada, e detalhes dos produtos devolvidos.

Também inclui informações associadas às vendas, como data da venda e ID do cliente.

Dentro de <returns>, cada <documentReturn> representa uma devolução, e há detalhes como ID da fatura, data de devolução, dias para devolução, se foi uma devolução antecipada, e detalhes dos produtos devolvidos.

A seção <associatedSales> inclui informações associadas às vendas, como data da venda e ID do cliente.

Estes ficheiros são colocados na pasta “webapp” localizada na paste de instalação do BaseX.

De seguida é aberto um servidor local a partir da execução do ficheiro “basexhttp.bat”, encontrado na pasta bin.

Utilização do Postman

Através do Postman executamos o pedido http ao Basex que por sua vez vai executar um pedido http com Query em MongoDB à Data API do Atlas, este por sua vez acede aos dados e retorna-os. A Data API do Atlas retorna os dados em JSON, o BaseX processa-os e retorna para o Postman os dados de acordo com o vocabulário estabilicido.

Para requisitar os dados referentes às vendas efetuamos um request GET introduzindo o seguinte URL: “http://localhost:<porta onde o servidor é acedido, por norma 8080 ou 8984>/getSales”, onde passamos como parâmetro o ano e o mês desejado.

Para requisitar os dados referentes às devoluções efetuamos um request GET introduzindo o seguinte URL: “http://localhost:<porta onde o servidor é acedido, por norma 8080 ou 8984>/getReturns”, onde, novamente, passamos como parâmetro o ano e o mês desejado. Foi enviado em anexo um ficheiro “.json” com a exportação do Postman.

Resumo e apreciação crítica

De uma forma geral o grupo sente que desenvolveu o trabalho pretendido.

A forma como todo o projeto foi planeado e realizado ao longo do tempo permitiu ao grupo desenvolver os seus conhecimentos e capacidades em relação ao domínio. Numa fase inicial estudamos e conhecemos a estrutura dos documentos fornecidos para depois podermos, de uma forma mais eficaz desenvolver um projeto que responde aos problemas propostos, através do vocabulário.

Ao longo do projeto algumas dúvidas foram surgindo, servindo como apoio ao estudo e à elaboração do projeto os slides divulgados na aula, bem como a documentação online das ferramentas utilizadas.

Bibliografia

- Moodle Processamento Estruturado de Informação.
- MongoDB documentation - https://www.mongodb.com/docs/
- BaseX documentation - https://docs.basex.org/wiki/Main_Page
- Postman documentation - https://learning.postman.com/docs/introduction/overview/