Processamento Estruturado de Informação

Trabalho Prático

Grupo 4 – 2º Entrega

José Baltar, 8170212

Rodrigo Coelho, 8170282

Rui Silva, 8170283



**Índice**

[1. Introdução 3](#_Toc530841496)

[1.1. Apresentação do Problema 3](#_Toc530841497)

[1.2. Contextualização 3](#_Toc530841498)

[2. Desenvolvimento do Projeto 4](#_Toc530841499)

[2.1. Estruturação e debate 4](#_Toc530841500)

[2.2. Requisitos 4](#_Toc530841501)

[2.3. Execução e explicação 5](#_Toc530841502)

[2.4. Propriedades dos *Schemas* 6](#_Toc530841503)

[2.5. Detalhes Finais 7](#_Toc530841504)

[3. Conclusão 8](#_Toc530841505)

# Introdução

## Apresentação do Problema

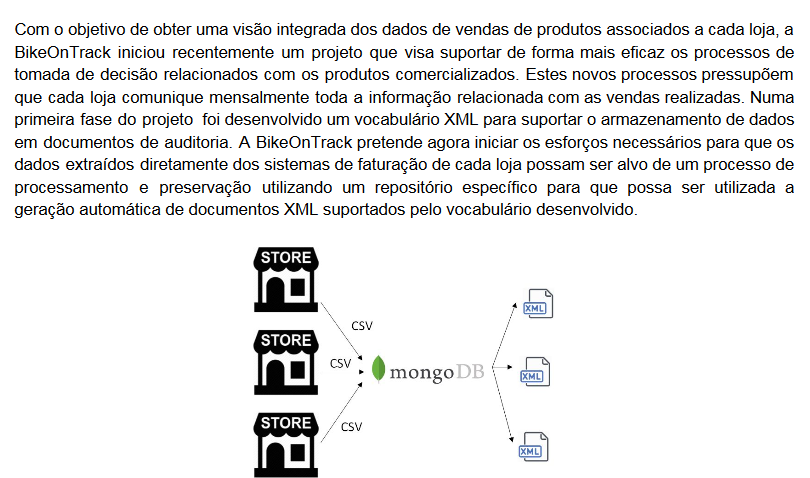
Para a Unidade Curricular de **Processamento Estruturado de Informação**, foi proposto, para Avaliação Contínua, a realização de um Trabalho Prático, divido em várias entregas. Para esta entrega, à semelhança da anterior, foi-nos colocado um enunciado com o problema para resolução, de forma a aplicar as diversas técnicas lecionadas.

Imagem Enunciado básico do problema

## Contextualização

Para desenvolver este projeto, foi necessário utilizar conhecimento adquirido nas aulas da U.C., para que se consiga adaptar o vocabulário aos novos problemas que são levantados, e tendo por base as resoluções e requisitos já anteriormente levantadas, e continuando com o cuidado de planear sempre para entregas futuras, de forma a ser possível um melhoramento contínuo e incremental do Trabalho.

Assim sendo, o projeto foi já pensado e planeado para permitir a melhor transformação possível dos documentos na base de dados em *MongoDB* para o vocabulário XML por nós criado na entrega anterior.

Este relatório servirá, por tantom como apoio à interpretação do projeto, de forma a elucidar em relação à estratégia e raciocínio seguidos, segundo a interpretação do enunciado apresentado, dúvidas e problemas encontrados.

# Desenvolvimento do Projeto

## Estruturação e debate

Inicialmente, tal como na anterior entrega, foi necessária a realização de um debate entre os membros do grupo face à abordagem a realizar, pois as interpretações do enunciado variavam de membro para membro. No entanto, sendo este enunciado mais explícito em relação ao anterior, foi mais fácil chegar a um consenso do que seria necessário desenvolver, e como o fazer.

Assim sendo, chegamos a três conclusões sobre o que seria necessário desenvolver:

* Adaptação dos Schemas e do nosso vocabulário aos novos requisitos no enunciado
* Criação de queries para MongoDB que permitissem moldar os dados para uma futura transformação em XML.
* Criação de um serviço REST, em Java, com o objetivo de implementar as queries desenvolvidas neste serviço, para apoiar na transformação para XML numa futura entrega.

## Requisitos

Com base nos requisitos anteriores,

## Execução e explicação

Com o projeto já planeado, foram divididas tarefas entre os membros para facilitar a sua execução, mas, uma vez que a comunicação e entreajuda são extremamente importantes para um projeto destes, os membros do grupo prontamente entraram e colaboraram no desenvolvimento das restantes componentes. Foi também utilizado o Git para ajudar na organização, desenvolvimento e comunicação.

As modificações feitas ao vocabulário procuram refletir as adições que o enunciado requere, e as queries executadas no MongoDB procuram criar os dados necessários para colmatar essas lacunas, que não se encontram nos documentos originais, em formato “csv”. O serviço REST criado procura ao máximo cobrir, com diversas funções, estas lacunas e executá-las, para mais tarde utilizar esses dados para a exportação para os ficheiros XML, através do Transformer.

Foi também necessário proceder à alteração da informação da coleção “ProductDetails”, que continha um problema com os ListPrices, em que algum dos documentos se encontravam com esse valor a 0, sendo necessário ir corrigir esses dados com base na collection “ProductListPriceHistory”, que continha os preços atuais dos produtos e as suas datas de utilização.

Assim sendo, as etapas de forma ordenada da elaboração do projeto foram:

* Interpretação do problema
* Debate sobre a abordagem a seguir e decisão final
* Atribuição e escalonamento de tarefas
* Execução das diversas tarefas atribuídas
* Revisão e correção de erros
* Elaboração do Relatório
* Entrega do projeto

## MongoDB

Seguindo os requisitos levantados do enunciado, foram desenvolvidas várias Queries para manipular os dados do *MongoDB,* como estas pretendiam.

Em primeiro lugar, foi necessário importar as collections (ou seja, os documentos csv) para o Mongo, através do comando “mongoimport”. Após termos as collections em base de dados, foram desenvolvidas queries básicas para perceber e relacionar corretamente os dados. Com isto, foram sendo estruturadas queries complexas, através da função “aggregate”. Adicionalmente, foi necessário recorrer a JavaScript para poder corrigir dados errados nas collections, nomeadamente o *ListPrice* da collection “ProductList”, em que alguns produtos se encontravam a 0, sendo necessário substituir pelo preço que se encontrava na “ProductListPriceHistory”. Nos requisitos, levantava-se o problema de não ser possível mudar o preço para diferente de 0, mas tal foi possível e nenhum dos produtos se encontra com o preço a zero.

As queries construídas refletem os requisitos levantados no enunciado da Entrega. Estas encontram-se no ficheiro “QueriesExecutadas.txt”, que é entregue com o projeto. As queries executadas

## Propriedades dos *Schemas (Alterar)*

Referindo-nos agora diretamente às componentes mais técnicas dos *Namespace* e *Schema* criados:

* ***CommonShema ("ProjetoPEI/Grupo/Entrega1/CommonSchema"):*** Com o *CommonSchema*, definimos os nossos tipos base que irão ser continuamente utilizados por todo o documento. Assim, evita-se a definição repetida de itens permitindo que todos estes tipos sejam utilizados facilmente. Adicionalmente, a existência deste *Schema* permite uma maior *reusabilidade* de todo o processo e melhor integração de novos componentes. Este *Schema* contém principalmente *simpleTypes* que possuem restrições de uso como padrões específicos de um determinado dado ou limite de valores, etc.
* ***SchemaLoja ("ProjetoPEI/Grupo/Entrega1/Loja"):*** Aqui é definido o *complexType* referente ao elemento “Loja” existente no *Schema* principal, tal como a definição de outros tipos específicos deste contexto, por exemplo o *complexType* “typeContactos” que disponibiliza, através do indicador de ordem “choice”, a possibilidade de introduzir 1 ou mais contactos (email ou telefone), devido aos indicadores de ocurrência “minOccurs(=”1”)” e “maxOccurs=(“unbounded”)”.
* ***SchemaCliente ("ProjetoPEI/Grupo/Entrega1/Cliente"):*** *Schema* idêntico ao anterior. Aqui está definido o *complexType* referente ao elemento “Cliente” existente no *Schema* principal, contendo elementos específicos do requisito “Cliente”, tal como um atributo que se refere ao ID ou número de Cliente.
* ***SchemaProduto ("ProjetoPEI/Grupo/Entrega1/Produto"):*** Neste *Schema* está estabelecido o *complexType* com elementos relativos ao elemento “Produto” a ser utilizado no vocabulário. A particularidade deste *Schema* é que vários elementos são “opcionais”, isto é o indicador de ocorrência “minOccurs” encontra-se igualado a 0. O vocabulário está escrito desta forma uma vez que essas informações não são relevantes para o contexto do problema (a documentação de vendas) bastando a existência da identificação do produto em questão.
* ***SchemaReceitas ("ProjetoPEI/Grupo/Entrega1/Receitas"):*** *Schema* alusivo ao componente de Vendas, mais propriamente o elemento “Receitas”. Neste caso são definidos os *complexTypes* respetivos à “Moeda” e “Receita”. No primeiro é delineado um *simpleType* com uma extensão de atributo que tem como base a enumeração definida no “CommonSchema” e um atributo opcional relativo ao ID da taxa de câmbio a ser utilizada. Relativamente ao segundo, estão definidos os elementos respeitantes ao requisito “Vendas e Receitas”, sendo criado globalmente um outro *complexType* relativo às linhas de Receita pertencentes à Receita original. Cada um desses tipos possui um atributo que representa a identificação dos mesmos.
* ***SchemaDocumentoGeral ("ProjetoPEI/Grupo/Entrega1/"):*** Este é o *Schema* que representa o Documento Geral e se encontram espelhados os requisitos do problema. Aqui é onde se encontra o *Root* da instância XML e a declaração dos elementos principais, tal como as datas respetivas dos mesmos presentes em atributo de uso obrigatório.

## Detalhes Finais

Diversas modificações ao projeto encontram num repositório *Git* que poderá ser encontrado em: “<https://github.com/JoseBaltar/ProjetoPEI>”. Os *commits* realizados ao repositório procuram representar aquilo que foi sendo feito por parte dos membros do grupo, mas não são réplicas fiéis do trabalho realizado por cada um. O esforço, sobretudo na procura de erros foi geral e discutida entre os membros do grupo, embora normalmente tenha sido apenas levado para o git por um membro. As queries, devido à sua natureza, foram sendo construídas separadamente do git, mas sempre em contacto constante entre os diversos membros do grupo.

A execução deste relatório, igualmente, recaiu na ponderação de todos os membros do grupo.

# Conclusão

Neste projeto abordamos o processo de adaptação e reutilização de um vocabulário XML, através da modificação de documentos XSD e *namespaces* associados, criação e estruturação de queries na plataforma *MongoDB*, e a utilização da API Java *Spring Rest*, de forma a planear a implementação das pesquisas para a transformação dos documentos em formato BSON, guardados na base de dados do *MongoDB,* para ficheiros XML, seguindo um conjunto de regras e requisitos.

Adicionalmente, uma importante componente deste projeto foi a nossa capacidade, enquanto estudantes e membros de um grupo de trabalho, na interpretação e discussão relativas a um problema pré-estabelecido, decidindo a abordagem a seguir.

Acreditamos que este projeto foi extremamente positivo para nós, pois permitiu-nos reforçar as nossas técnicas e conhecimentos relativos aos conteúdos usados, incluindo na criação deste relatório. Não apenas isto, mas também foi essencial para a formação de um maior espírito crítico em relação à análise e abordagens de problemas práticos.