

# **Relatório de Processamento Estruturado de Informação**

## **Grupo 13**

8210414 – Gil leão

8210389 – Paulo Gonçalves

## Índice

1 - Objetivo .....	3
2 – Vocabulário XML .....	3
3 – REST API.....	3
4 – Conclusão .....	4

## 1 - Objetivo

Este trabalho tem como objetivo o desenvolvimento de um projeto, uma empresa que tem como principal atividade a realização de inspeções mecânicas a automóveis. Esta empresa possui vários peritos e parceiros para assegurar que todas as peritagens são realizadas. Para facilitar o envio da informação relacionada com as inspeções realizadas por cada parceiro, foi necessária a criação de um vocabulário XML e de uma REST API.

## 2 – Vocabulário XML

Para este projeto foi necessária a criação de um vocabulário XML que contempla-se as seguintes informações: a informação sobre o conjunto de peritagens realizadas num determinado dia por um perito e, a informação sobre um conjunto de marcações de peritagens num determinado dia por um perito. Decidimos também criar e adicionar à base de dados os peritos e os parceiros manualmente.

## 3 – REST API

Utilizamos uma REST API, mais especificamente o módulo RESTXQ disponibilizado pelo BaseX, de forma a possibilitar a comunicação entre o BaseX e o Postman através de pedidos HTTP.

A RESTXQ é uma API que permite a utilização do XQuery como linguagem server-side para a web. Esta define um conjunto de anotações pré-definidas que permitem o mapeamento de pedidos HTTP em funções XQuery que geram e retornam respostas HTTP.

Para este projeto, usamos funções XQuery que permitem:

- Validar o documento XML submetido (função `validateXMLmarcacao` e função `validateXMLperitagem`);
- Validar se o código de marcação, parceiro e perito existem (função `addPeritagem` e função `addMarcacao`);
- Substituir e um documento correspondente às peritagens de um determinado dia (função `alterar`);
- Apresentar as marcações associadas a um perito para o dia atual (função `findMarcacoes`);
- Apresentar as peritagens realizadas/não realizadas num determinado dia para um dado parceiro (função `seePeritagens`);
- Consultar os dados de uma peritagem (função `seePeritagem`) utilizando o código associado às peritagens como argumento de validação;
- E por fim, adicionar peritagens e marcações (função `addPeritagem2` e função `addMarcacao2`).

## 4 – Conclusão

Com este trabalho foi possível melhorar e consolidar o nosso conhecimento e as nossas capacidades em relação à matéria, neste caso, relativamente à criação de vocabulários XML, à criação do XML Schema, da criação de bases de dados no BaseX e o funcionamento de pedidos HTTP com o Postman.

Relativamente a dificuldades que surgiram neste trabalho, surgiu um problema em algumas funções na REST API, que impossibilitaram o seu correto funcionamento, que depois de várias tentativas para ser resolvido, mesmo com a ajuda do professor das aulas práticas, não se encontrou o erro no projeto (suspeitando-se do XML).