ENGENHARIA INFORMÁTICA
Enunciado do trabalho prático n.1

Integração de Sistemas (de Informação)

Ano letivo 2023/2024

Objetivo

O presente trabalho prático 1 (TP1) tem como objetivo desenvolver as capacidades dos estudantes na utilização de métodos de representação de dados (através da utilização de linguagens de anotação) para auxiliar a integração de diferentes sistemas e permitir assim a interoperabilidade dos mesmos.

Leia atentamente todo o enunciado e contacte os docentes da disciplina em caso de dúvidas.

- Cada **grupo de trabalho** deverá ter **2 ou 3 alunos, inscritos no mesmo turno.** Trabalhos submetidos por somente 1 aluno só serão permitidos em casos excecionais (ex.: trabalhador estudante com o estatuto aprovado), que deverão ser discutidos previamente com os docentes da disciplina.
- Os alunos deverão registar o grupo de trabalho no Moodle, indicando o subtema escolhido. **A não inscrição atempada do grupo** de trabalho no Moodle irá ser interpretada como **uma desistência da realização de avaliação contínua** nesta disciplina.

Regras

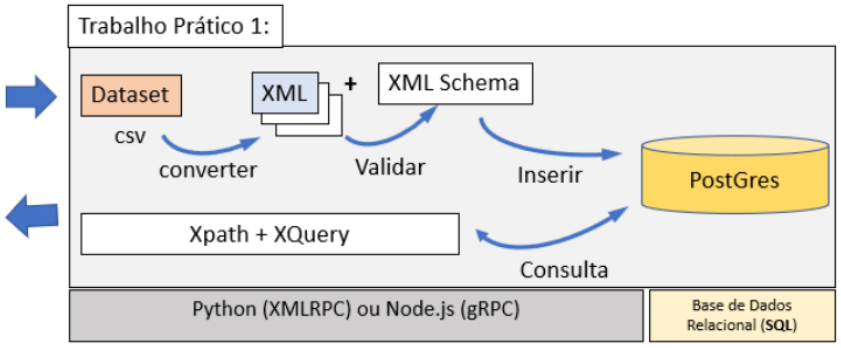
- Para o desenvolvimento dos trabalhos propostos, deverá ser usada a linguagem de programação Python.
- Os pormenores de implementação que se devem à interpretação dos enunciados por parte dos grupos de alunos deverão ser descritos no relatório com detalhe e justificação das opções tomadas.
- A implementação de funcionalidades extra não presentes no enunciado será valorizada, desde que estas funcionalidades não modifiquem os requisitos obrigatórios e não reduzam a dificuldade do trabalho. As funcionalidades extra implementadas deverão ser documentadas no relatório.
- A apresentação de relatórios e/ou implementações **não originais e que constituam plágio**, conduzem à imediata atribuição de **nota zero** no trabalho de grupo e a **eventuais processos disciplinares**.

Avaliação e Entrega

- O trabalho prático 1 faz parte da avaliação da Componente Prática da disciplina de **Integração de Sistemas** (correspondendo a 50% da nota final).
- A nota é atribuída individualmente aos elementos do grupo segundo a apresentação, visualização e discussão dos elementos entregues e as impressões obtidas pelos docentes acerca do aluno durante o decorrer das aulas de acompanhamento.
- Para aprovação à disciplina, a nota da Componente Prática deverá ter a classificação mínima obrigatória de **10.0 valores**.
- As discussões orais deste trabalho serão realizadas no dia indicado no calendário e no horário previamente definido para o efeito. Os grupos de trabalho devem reservar o dia e hora marcados, na sua agenda. Os trabalhadores-estudantes devem, ao abrigo do respetivo estatuto, solicitar a folha justificativa de exame para a empresa.
- O trabalho prático deverá ser submetido através do Moodle seguindo as instruções lá indicadas. A entrega deverá conter os seguintes elementos:
    - Código fonte em Python;
    - Slides em PowerPoint / PDF (seguindo estrutura a indicar);
    - Vídeo com duração máxima de 5 minutos com instruções de utilização e demonstração de funcionalidades;

Descrição do trabalho

Esquemático Geral:

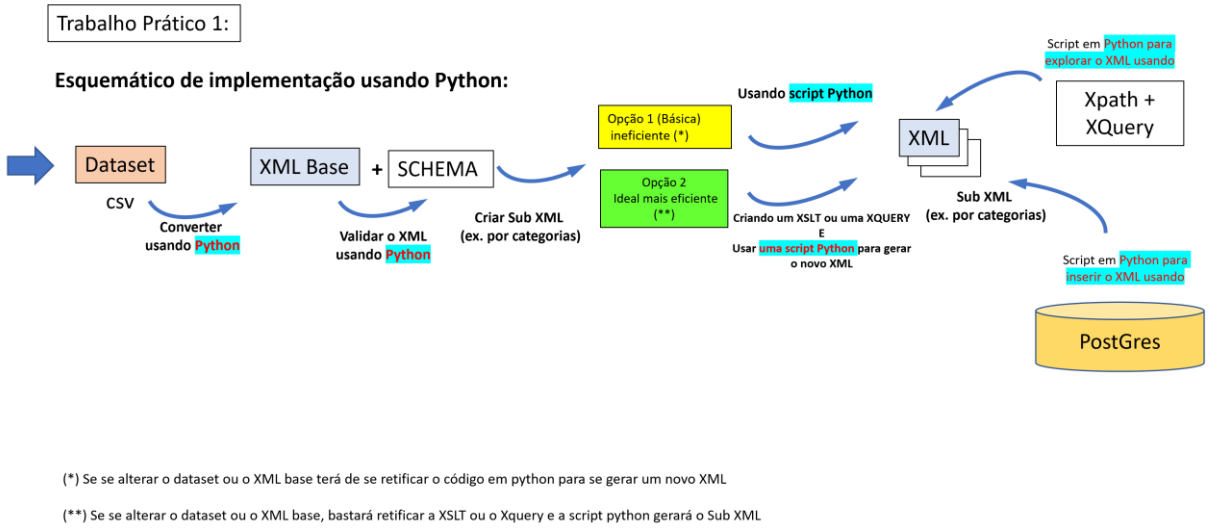


1. Para o trabalho é disponibilizada a base de código em um repositório git, que deverá ser utilizada nos trabalhos. A base de código contém uma configuração em Docker Compose com todas as dependências do projeto, assim como instruções de utilização. Os scripts podem ser editados caso os alunos achem relevante adicionar alguma nova dependência.
2. Cada grupo deverá escolher um dataset em https://www.kaggle.com/datasets. O dataset escolhido deverá ter as seguintes características:
    a. Não são permitidos datasets em XML
    b. Deverá ser único para cada grupo do mesmo turno
    c. Deverá ter pelo menos 10000 registos
    d. Deverão ser privilegiados datasets com campos nominais e não somente numéricos. (ver exemplo dataset menos adequado).
    e. O dataset deverá ter algum campo que permita identificar a localização do registo. Alguns exemplos:
        i. NBA Players: O campo "college" permite identificar onde os jogadores estudaram.
        ii. FIFA 23 Complete Player Database: O campo "Country" pode ajudar a identificar onde o jogador nasceu e o campo "Club" podem ajudar a localizar onde o jogador joga.
3. Os alunos deverão comunicar aos docentes através do Moodle o dataset escolhido. Se 2 grupos escolherem o mesmo dataset, a atribuição será por ordem de chegada.
4. O grupo deverá definir um **novo formato XML** para guardar os dados do dataset escolhido. A definição do formato XML deverá ser acompanhada pelo respetivo XML Schema, que pode ser usado para validar documentos no novo formato.
    a. O formato do XML deverá ser criado pelos alunos. Deverão ser seguidas as boas práticas no que toca à criação de formatos XML.
    b. O novo formato de dados deverá tirar partido das características hierárquicas do XML. Isto significa que não serão aceites documentos XML que fazem uma simples conversão das colunas do CSV para um ficheiro XML *flat*.
    c. No caso do Schema, deverá ser realizada uma validação extensiva de todos os campos do documento. Exemplos: validação dos tipos de dados e validação de chaves únicas ou referências cruzadas, entre outros.

5. Deverá ser criado um servidor XMLRPC em Python. Todos os métodos de importação ou de perguntas à base de dados deverão ser disponibilizados através deste servidor.
6. Deverá ser criado um método que converte dados do dataset em CSV para o novo formato de XML.
7. O novo formato XML, além de conter os dados do dataset em CSV, deverá conter informação adicional relativa aos campos de localização, nomeadamente as coordenadas GPS dessa mesmo localização.
    a. Os dados podem ser obtidos usando a Search API do Nominatim, usando o módulo de HTTP Requests já existente no Python.
8. Deverá ser criado um método para importar documentos XML para uma base de dados PostgreSQL. Os documentos deverão ser importados para uma tabela única com os seguintes campos:
    a. ID (único auto-gerado)
    b. Nome do ficheiro importado (VARCHAR)
    c. Conteúdo do ficheiro importado em XML (tipo XML)
    d. Data de importação (datetime)
9. Deverá ser possível remover ou adicionar ficheiros novos através de XMLRPC. No caso de remoção, deverá ser feito um "*soft-delete*".
10. Deverão ser implementadas pelo menos 5 rotinas RPC para consultas de dados nas colunas XML da base de dados construída (exemplo: se a base de dados for uma biblioteca, uma rotina exemplo seria a consulta de todos os livros publicados por determinado autor no 1987).
    a. As consultas de deverão usar XPATH/XQuery sempre que possível para otimizar a busca de resultados
    b. As consultas serão realizadas a todos os ficheiros importados. Deverá ser indicada na chamada à função RPC se os resultados deverão ser agrupados por ficheiro ou acumulados num único resultado.
    c. As rotinas deverão retornar somente os dados essenciais (ao invés de retornar objetos completos)
    d. Deverão ser escolhidas rotinas com algum nível de complexidade:

i. Incluir pesquisa em texto;
ii. Inclusão de filtros;
iii. Agrupamento de resultados;
iv. Ordenação de resultados;
v. Intercambio de informação entre diversos níveis do documento.

Uma proposta/sugestão de dois possíveis caminhos para a implementação é apresentada na figura seguinte, usando apenas implementação em Python ou usando a linguagem de programação Python e recorrendo a scripts XSLT e/ou XQUERY:



(*) Se se alterar o dataset ou o XML base terá de se retificar o código em python para se gerar um novo XML

(**) Se se alterar o dataset ou o XML base, bastará retificar a XSLT ou o Xquery e a script python gerará o Sub XML

Salienta-se que **não devem elaborar formatos XML** em "forma de tabela" (exemplo abaixo), isto é, de modo a que aproveitem ao máximo as características da linguagem para melhor representar a informação (usando as hierarquias do formato).

```
<tabela>
   <linha>
       <col_a> valor_a </col_a>
       <col_b> valor_b </col_b>
       ....
   </linha>
   <linha>
       <col_a> valor_a </col_a>
       <col_b> valor_b </col_b>
       ....
   </linha>
   ...
</tabela>
```