

# Biblioteca de Manipulação de XML

Projeto de Programação Avançada 2023/2024

Pretende-se uma biblioteca para geração e manipulação de [XML](#) em Kotlin, cobrindo os elementos essenciais:

- Documento
- Entidades (*tags*)
- Texto aninhado em entidade
- Entidades aninhadas
- Atributos

O seguinte exemplo ilustra os elementos acima.

```
<?xml version="1.0" encoding="UTF-8"?>
<plano>
  <curso>Mestrado em Engenharia Informática</curso>
  <fuc codigo="M4310">
    <nome>Programação Avançada</nome>
    <ects>6.0</ects>
    <avaliacao>
      <componente nome="Quizzes" peso="20%"/>
      <componente nome="Projeto" peso="80%"/>
    </avaliacao>
  </fuc>
  <fuc codigo="03782">
    <nome>Dissertação</nome>
    <ects>42.0</ects>
    <avaliacao>
      <componente nome="Dissertação" peso="60%"/>
      <componente nome="Apresentação" peso="20%"/>
      <componente nome="Discussão" peso="20%"/>
    </avaliacao>
  </fuc>
</plano>
```

O objetivo é assumir o papel de quem está a desenvolver algo para terceiros utilizarem (biblioteca). Assim sendo, aspetos de API (forma e documentação) deverão ser tidos em especial atenção. Por outro lado, a biblioteca deverá oferecer alguma flexibilidade na sua utilização para permitir aplicá-la num leque alargado de situações.

**Atenção:** A biblioteca não tratará de *parsing* de XML (embora isso fizesse sentido numa situação real). Apenas é exigida a geração de XML a partir de modelos em memória.

**Documentação:** A API da biblioteca deverá estar documentada utilizando [KDoc](#).

**Testes:** Todas as funcionalidades deverão ter os respectivos testes [JUnit](#).

# Fase 1 - Modelo

**Objetivo:** Desenvolver as classes para representar XML em memória (modelo), operações de manipulação, e testes.

## API

As classes deverão permitir:

1. **adicionar e remover entidades**
2. **adicionar, remover, e alterar** atributos em entidades
3. **aceder à entidade mãe e entidades filhas** de uma entidade
4. ***pretty print*** em formato de String, e **escrita para ficheiro**
5. varrimento do documento com objetos **visitantes** (*Visitor*)
6. **adicionar atributos** globalmente ao documento  
(fornecendo *nome* da entidade, *nome* e *valor* do atributo)
7. **renomeação de entidades** globalmente ao documento  
(fornecendo *nome antigo* e *nome novo*)
8. **renomeação de atributos** globalmente ao documento  
(fornecendo *nome* da entidade, e *nome antigo* e *nome novo* do atributo)
9. **remoção de entidades** globalmente ao documento  
(fornecendo *nome*)
10. **remoção de atributos** globalmente ao documento  
(fornecendo *nome* da entidade, e *nome* de atributo)

## Micro-XPath

Com vista a ter uma forma de obter fragmentos de XML, pretende-se permitir interrogar o documento com expressões [XPath](#) *simples*. Entenda-se, expressões apenas compostas por uma sequência de nomes de entidades. Por exemplo a expressão seguinte resultaria na seguinte lista de entidades:

*fuc/avaliacao/componente*

```
<componente nome="Quizzes" peso="20%"/>
<componente nome="Projeto" peso="80%"/>
<componente nome="Dissertação" peso="60%"/>
<componente nome="Apresentação" peso="20%"/>
<componente nome="Discussão" peso="20%"/>
```