

# MAC122 Princípios de Desenvolvimento de Algoritmos

## EP no. 2

Prof. Dr. Paulo Miranda  
Instituto de Matemática e Estatística (IME)  
Universidade de São Paulo (USP)

O XML (*eXtensible Markup Language*) é um formato recomendado pela W3C para a criação de documentos com dados organizados de forma hierárquica, cujo propósito principal é facilitar o compartilhamento de informações através da internet.

Um documento em XML é um texto que apresenta marcações, chamadas tags (etiquetas), que são usadas para organizar o conteúdo do documento. Tags são identificadas como trechos do texto compreendidos entre um caracter de início '`<`' e outro caracter de fim '`>`'. Genericamente falando, existem dois tipos de tags - tags de abertura: `<comando>` e tags de fechamento: `</comando>`. A diferença entre elas é que na tag de fechamento existe uma barra '`/`'. Abaixo é mostrado um exemplo de documento em XML:

```
<?xml version="1.0"?>
<note>
  <to>Jane</to>
  <from>Tarzan</from>
  <heading>Reminder</heading>
  <body>Jane must stay with Tarzan!</body>
</note>
```

Você deve ter notado a partir do exemplo anterior que a declaração `<?xml ... ?>` da primeira linha não tem uma tag de fechamento correspondente. Isso não é um erro. Essa primeira declaração não tem por padrão nenhuma marca de fechamento. Ela apenas informa a versão do documento XML.

As tags de abertura podem possuir atributos na forma de pares `nome="valor"`. Exemplo:

```
<?xml version="1.0"?>
<note date="12/11/2007">
  <to>Jane</to>
  <from>Tarzan</from>
</note>
```

No exemplo acima `date="12/11/2007"` é um atributo da tag `note`.

Um documento em XML apresenta as seguintes regras de sintaxe:

1. Os documentos XML são sensíveis à letras maiúsculas e minúsculas.  
Ex: `<to>Jane </T0>` está errado.

2. Todos os elementos XML devem ter uma tag de fechamento.

Exemplo de documento incorreto:

```
<?xml version="1.0"?>
<note date="12/11/2007">
  <to>Jane
  <from>Tarzan
</note>
```

3. Todos os elementos XML devem estar devidamente aninhados.

Exemplo de documento incorreto:

```
<?xml version="1.0"?>
<note date="12/11/2007">
  <to>Jane</note>
  <from>Tarzan</from>
</to>
```

4. Todos os documentos XML devem ter um elemento raiz que contém todos os outros elementos.  
Por exemplo, o elemento raiz dos exemplos acima é a tag `<note>`.

Para um dado arquivo texto, contendo um documento no formato XML, faça um programa que verifica se as tags de abertura e fechamento estão balanceadas. O seu programa deve imprimir uma mensagem de erro caso alguma das regras de 1 a 4 for violada. Para a resolução do problema, utilize uma implementação de pilha com listas ligadas. Exemplo de documento mais complexo com tags balanceadas:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<receita nome="pao" tempo_de_preparo="5 minutos" tempo_de_cozimento="1 hora">
  <titulo>Pao simples</titulo>
  <ingredientes>
    <ingrediente quantidade="3" unidade="xicaras">Farinha</ingrediente>
    <ingrediente quantidade="7" unidade="gramas">Fermento</ingrediente>
    <ingrediente quantidade="1.5" unidade="xicaras" estado="morna">Agua</ingrediente>
    <ingrediente quantidade="1" unidade="colheres de cha">Sal</ingrediente>
  </ingredientes>
  <instrucoes>
    <passo>Misture todos os ingredientes, e dissolva bem.</passo>
    <passo>Cubra com um pano e deixe por uma hora em um local morno.</passo>
    <passo>Misture novamente, coloque numa bandeja e asse num forno.</passo>
  </instrucoes>
</receita>
```

### **Observações:**

**Note que a indentação (reco) é geralmente usada em arquivos XML, pois torna o documento mais legível para humanos. No entanto, ela não é necessária em documentos XML. O seu programa não precisa verificar questões relativas a indentação do documento.**

**Assuma que os caracteres '<' e '>' sempre aparecem aos pares e nessa ordem (ou seja, para todo caracter '<' o próximo caracter de marcação é sempre um '>').**

**O nome do elemento deve seguir imediatamente a marca de início '<'. Isso significa que não há espaços em branco entre o nome e '<'. Se você estiver criando um elemento raiz chamado de message, ele deve ler: <message> e não < message>. O seu programa pode assumir que os elementos foram digitados corretamente.**

**Por questões de simplicidade, assumo o padrão ASCII.**

**Algumas funções para a manipulação de arquivos como a função getc, rewind, fseek, e ftell podem ser úteis. Consulte a documentação dessas funções na internet ou em livros especializados.**