

Documentação complementar:

- [Prolog XML](#)
- [XML Encoding](#)
- [JAXP](#)

Parte 1

1. Considere o seguinte documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<ingredient>
  <name>Tomato</name>
  <category>Vegetal</category>
  <unit>grams</unit>
  <calories>18</calories>
</ingredient>
```

1.1. Altere a estrutura de forma a acomodar o armazenamento de dados relativos aos seguintes ingredientes:

Nome	Category	Unit	Calories/100g
Spinach	Vegetal	grams	23
Cheese	Dairy	grams	300
Tomato Sauce	Sauce	milliliters	18
Oregano	Herb	grams	265
Chicken	Meat	grams	239
Onion Sauce	Sauce	milliliters	92
Pasta	Pasta	grams	131

Inicie a resolução do exercício pela elaboração de uma árvore representativa da organização do documento XML.

1.2. Altere a estrutura de forma a incluir uma descrição do ingrediente e ainda o tipo do ingrediente quando inserido no contexto de uma pizza: "Topping" or "Base";

2. Considere o seguinte documento que representa um conjunto de dados relacionados com a ementa de um restaurante:

A TASCA DA ESTG	
Menu	
Veggie-Pizza (VEP)	6.00€ (S)
Base: tiny	8.00€ (M)
Topping: spinach, cheese, and tomato sauce	12.00€ (B)
Mozzarella-Pizza (MZP)	7.00€ (S)
Base: crispy	8.50€ (M)
Topping: cheese, oregano, and onion sauce	13.00€ (B)
Meat-Pizza (MP)	6.00€ (S)
Base: lean	9.00€ (M)
Topping: chicken, oregano and tomato sauce	12.50€ (B)
ESTG-Pizza (ESTGP)	6.00€ (S)
Base: tiny	8.50€ (M)
Topping: spinach, cheese, tomato sauce, and chicken	15.00€ (B)

Estruture os dados apresentados num documento XML e verifique se o documento criado se encontra formado corretamente.

Parte 2

1. Considere o documento XML resolvido na parte 1. Utilizando a biblioteca JAXP¹ da linguagem Java, desenvolva um pequeno programa que utilizando a API DOM, realize a leitura do ficheiro XML e apresente o seu conteúdo na linha de comandos.

Resolução parcial

Após criar um projeto Java, crie uma classe: *GestaoXML* para suportar a leitura de um documento XML.

Importação das classes necessárias:

```
import java.io.File;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
```

¹ <https://docs.oracle.com/javase/tutorial/jaxp/index.html>

Criar uma classe para realizar a gestão do documento XML:

```
public class GestaoXML {  
  
    private final String filename;  
    private Document doc;  
  
    /**  
     * Método construtor que permite a instânciação da classe com o  
     * nome/localização do documento XML a processar  
     */  
    /**  
     * @param filename Caminho/nome do documento XML a processar  
     */  
    public GestaoXML(String filename) {  
        this.filename = filename;  
    }  
  
    (...)
```

Criar um método para suportar a leitura do documento XML para memória:

```
/**  
 * Método responsável por ler para memória o documento XML  
 */  
public void readXML() {  
    //Invocação e instanciação dos objetos que suportam a leitura do documento XML  
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();  
    DocumentBuilder db;  
    try {  
        //Define a API para obter instâncias do documento XML, permitindo extrair o documento do ficheiro XML.  
        db = dbf.newDocumentBuilder();  
        //invocação do parser para processamento do documento  
        doc = db.parse(new File(filename));  
    } catch (ParserConfigurationException | SAXException | IOException ex) {  
        Logger.getLogger(GestaoXML.class.getName()).log(Level.SEVERE, null, ex);  
    }  
  
}
```

Criar um método para imprimir o conteúdo do documento:

```
public void printMenuXML() {  
    //Permite devolver o elemento que se encontra no root do documento  
    System.out.println("Menu :" + doc.getDocumentElement().getNodeName());  
    //permite retornar lista de elementos do documento que são filhos do elemento root  
    NodeList pizzas = doc.getChildNodes().item(0).getChildNodes();  
    for (int i = 0; i < pizzas.getLength(); i++) {  
        //aceder a filhos do tipo nodo  
        if (pizzas.item(i).getNodeType() == Node.ELEMENT_NODE) {  
            Element pizzaElement = (Element) pizzas.item(i);  
            //listagem dos elementos  
            System.out.println("Pizza cod: " + pizzaElement.getElementsByTagName("codigo").item(0).  
                getTextContent());  
        }  
    }  
  
    (...)
```

Criar ou utilizar a classe main existente:

```
public class PEI1718FP1 {  
  
    public static void main(String[] args) {  
        //Instanciar a classe responsável pela gestão do XML  
        GestaoXML meuGestor = new GestaoXML("menu.xml");  
        //Invocar o método responsável por ler o documento XML  
        meuGestor.readXML();  
        //Invocar o método responsável por imprimir o conteúdo do documento XML  
        meuGestor.printMenuXML();  
    }  
  
}
```

2. Com recurso à biblioteca JAXP, implemente uma nova versão do exercício anterior para suportar a leitura e impressão de dados de qualquer documento XML, independentemente da sua estrutura (implemente o código de forma a percorrer toda a árvore XML).