

02

## Outra forma de ler um XML

### Transcrição

Para não perdemos o código que construímos até agora na classe `Sistema.java`, criaremos uma nova classe `LeArquivoXmlDeOutraForma.java` - um nome meio grande, mas no futuro pensaremos em um melhor. Se essa classe será executada pelo console, precisaremos de um método `main()`. Como usaremos outra estratégia para ler o arquivo XML, não usaremos a `DocumentBuilderFactory`, mas sim outras classes que implementam tal estratégia.

Começaremos pedindo para a classe `XMLReaderFactory` uma instância de `XMLReader` por meio do método `createXMLReader()`, e atribuiremos tal instância a uma variável `leitor`. Faremos as importações necessárias e adicionaremos um `throws Exception` para lidar com as exceções checadas do método `createXMLReader()`.

```
public class LeArquivoXmlDeOutraForma {
    public static void main(String[] args) throws Exception {
        XMLReader leitor = XMLReaderFactory.createXMLReader();
    }
}
```

A partir do `leitor`, chamaremos o método `parse()` que nos permitirá ler um arquivo. Esse método pede um *input source*, que chamaremos de `is` e criaremos a respectiva variável local. Criaremos então uma instância de `InputSource()` que, por sua vez, pede um *input stream*, que chamaremos de `ips`. Essa variável receberá uma instância de `FileInputStream()`, que por sua vez receberá o caminho do nosso arquivo XML - ou seja, `src/vendas.xml`.

```
public class LeArquivoXmlDeOutraForma {
    public static void main(String[] args) throws Exception {
        XMLReader leitor = XMLReaderFactory.createXMLReader();
        InputStream ips = new FileInputStream("src/vendas.xml");
        InputSource is = new InputSource(ips);
        leitor.parse(is);
    }
}
```

Se executarmos o código dessa forma, não teremos retorno algum, afinal não construímos nenhuma lógica que nos devolva qualquer informação. Queremos que o nosso `leitor` execute um bloco de código, e no Java isso pode ser feito por meio de um método ou de uma classe. No nosso caso, criaremos uma classe `LeitorXml.java`. A partir do `leitor`, chamaremos o método `setContentHandler()`.

```
public class LeArquivoXmlDeOutraForma {
    public static void main(String[] args) throws Exception {
        XMLReader leitor = XMLReaderFactory.createXMLReader();
        leitor.setContentHandler(arg0);
        InputStream ips = new FileInputStream("src/vendas.xml");
        InputSource is = new InputSource(ips);
        leitor.parse(is);
    }
}
```

Para executarmos a lógica de `LeitorXml.java`, essa classe precisará ser uma instância de `ContentHandler`. Para isso, como estamos trabalhando com orientação a objetos, usaremos a herança de `DefaultHandler`.

```
public class LeitorXml extends DefaultHandler{  
  
}
```

Assim, criaremos uma variável `logica`, que será uma instância de `LeitorXml()`, e a passaremos para o método `setContentHandler()`.

```
public class LeArquivoXmlDeOutraForma {  
    public static void main(String[] args) throws Exception {  
        XMLReader leitor = XMLReaderFactory.createXMLReader();  
        LeitorXml logica = new LeitorXml();  
        leitor.setContentHandler(logica);  
        InputStream ips = new FileInputStream("src/vendas.xml");  
        InputSource is = new InputSource(ips);  
        leitor.parse(is );  
    }  
}
```

No próximo vídeo aprenderemos a executar cada método do nosso `leitor` e a ler o arquivo XML.