



Instituto Politécnico
de Viana do Castelo



Instituto Politécnico
de Viana do Castelo

GRADUATION IN INFORMATICS ENGINEERING



Escola Superior
de Tecnologia e Gestão

Practical Work nº1

Parsing of XML Files using REST and
SOAP via JAVA, PHP and PYTHON

■ Course Unit: Integração de Sistemas

Eduardo Eiras Nº 21484

■ Coordinator: Prof. Doctor Jorge Ribeiro

Instituto Politécnico de Viana do Castelo
Escola Superior de Tecnologia e Gestão
www.ipvc.pt

■ Summary

1. Introduction and Objectives

2. Parsers XML in JAVA

2.1. Parser DOM

2.2. Parser SAX

2.3. Parser JDOM

2.4. Parser JAXP

2.5. Parser JAXB

3. SOAP and REST WebServices Implementation integrating XML files in a Data Base

3.1. Java webservices implementation

3.1.1. SOAP webservice with file books.xml and JAXB parser

3.1.2. REST webservice with file books.xml and JAXB parser

3.1.3. SOAP webservice with file simple.xml and JAXB parser

3.1.4. REST webservice with file simple.xml and JAXB parser

3.1.5. SOAP webservice with file alunos.xml from database

3.1.6. REST webservice with file alunos.xml from database

■ Summary

3.2. PHP webservices implementation

- 3.2.1. REST webservice to read data form the database
- 3.2.2. REST webservice using parser DOM/SAX and the books.xml file
- 3.2.3. SOAP webservice to read data from the database
- 3.2.4. SOAP webservice to read data from the file books.xml using SAX parser

4. Output correction in client side

- 4.1. REST webservice with the file books.xml and parser DOM
- 4.2. SOAP webservice with the file books.xml and parser SAX

5. Publish a Java project in a Tomcat server

6. XPATH exploration in XML files

- 6.1. REST webservice with the file books.xml and parser JAXB - Java
- 6.2. SOAP webservice with the file books.xml and parser DOM - PHP

■ Summary

7. Submitting large XML files

- 7.1. REST webservice and parser JAXB - Java
- 7.2. SOAP webservice and parser DOM – PHP
- 7.3. Problems and approaches to process large files

8. SOAP and REST web services implementation in Python and JavaScript

- 8.1. Python – REST web service
- 8.2. Python – SOAP web service
- 8.3. JavaScript – REST web service
- 8.4. JavaScript – SOAP web service

9. Python SOAP and REST web services to read data from MySQL database

- 9.1. Python – REST web service
- 9.2. Python – SOAP web service

10. Conclusion

11. Bibliography and Web References

■ 1. Introduction and Objectives

■ The present document demonstrates the work done, following the proposal of the first practical work which includes, in the first part, the execution of tutorial number 8, using different parsers with different XML files, both in structure and size, implementing concepts already learnt from the execution of tutorial number 7, as well as the implementation of SOAP and REST web services using the client/server approach and different XML files. Additionally, the implementation of web services in tutorial number 8 is done in two different languages which are JAVA and PHP.

The second part of this practical work, aims to approach and use XPATH syntax to filter data in XML files, creating web services in Python and JavaScript, accessing a MySQL database using Python, uploading big XML files to a server and publish a java project to a Tomcat server.

■ 2. Parsers XML in JAVA

■ There are several different XML parsers, these can be used to de-serialize XML files that serve as their input. This section demonstrates the use of five different parsers those being:

- DOM;
- SAX
- JDOM
- JAXP
- JAXB

These will be used to de-serialize the input of three different sized files sent from a client to a server. The demonstration of this execution will be done locally, using four CMD windows, one running the server, the second running the JavaRPC, the third the client and the fourth the output of the file received and parsed by the server. The window running the JavaRPC is only going to be presented in the first test, the reason for this is that this window's content is the same for all the tests and it is necessary to be always running to ensure that the connection between the server and the client is successful.

■ 2. Parsers XML in JAVA

- Additionally for the tests involving the files medsamp2013.xml and medsamp2014.xml, it was necessary to correct the structure of both XML files, in order to be able to parse them correctly, this because there were several tags missing at the end of the files.

■ 2.1. Parser DOM

Integração de Sistemas

■ simple.xml

```
C:\Windows\system32\cmd.exe
C:\TESTAR>set CLASSPATH=%CLASSPATH%;C:\TESTAR\orpc.dev\lib\orpc.jar
C:\TESTAR>java FileClient localhost simple.xml
Error: Could not find or load main class FileClient
Caused by: java.lang.ClassNotFoundException: FileClient

C:\TESTAR>set CLASSPATH=%CLASSPATH%;.

C:\TESTAR>java FileClient localhost simple.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
simple.xml 1135 bytes enviados
```

File Client

```
C:\Windows\system32\cmd.exe - java FileServer
C:\TESTAR>java FileServer
Error: Could not find or load main class FileServer
Caused by: java.lang.ClassNotFoundException: FileServer

C:\TESTAR>set CLASSPATH=%CLASSPATH%;.

C:\TESTAR>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: simple.xml 1135 bytes transferidos
Ficheiro gravado: simple.xml
Closing socket.
```

File Server

```
DOC: nodeName="#document"
COMM: nodeName="#comment" nodeValue=" Edited by XMLSpy® "
ELEM: nodeName="breakfast_menu" localName="breakfast_menu"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="food" localName="food"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="name" localName="name"
TEXT: nodeName="#text" nodeValue="Belgian Waffles"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="price" localName="price"
TEXT: nodeName="#text" nodeValue="$5.95"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="description" localName="description"
TEXT: nodeName="#text" nodeValue="Two of our famous Belgian Waffles with plenty of real maple syrup"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="calories" localName="calories"
TEXT: nodeName="#text" nodeValue="650"
TEXT: nodeName="#text" nodeValue=[WS]
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="food" localName="food"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="name" localName="name"
TEXT: nodeName="#text" nodeValue="Strawberry Belgian Waffles"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="price" localName="price"
TEXT: nodeName="#text" nodeValue="$7.95"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="description" localName="description"
TEXT: nodeName="#text" nodeValue="Light Belgian waffles covered with strawberries and whipped cream"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="calories" localName="calories"
TEXT: nodeName="#text" nodeValue="900"
TEXT: nodeName="#text" nodeValue=[WS]
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="food" localName="food"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="name" localName="name"
TEXT: nodeName="#text" nodeValue="Berry-Berry Belgian Waffles"
```

Output

```
C:\Windows\system32\cmd.exe - java -cp ".;orpc.jar" pmapsvc
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\TESTAR>set CLASSPATH=%CLASSPATH%;C:\TESTAR\orpc.dev\lib\orpc.jar
C:\TESTAR>java -cp ".;orpc.jar" pmapsvc
Netbula JavaRPC demo, not distributable!
```

Portmapper

■ 2.1. Parser DOM

books.xml

```
C:\Program Files\Common Files\Oracle\Java\javapath\java.exe
DOC: nodeName="#document"
ELEM: nodeName="catalog" local="catalog"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="book" local="book"
ATTR: nodeName="id" local="id" nodeValue="bk101"
TEXT: nodeName="#text" nodeValue="bk101"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="author" local="author"
TEXT: nodeName="#text" nodeValue="Gambardella, Matthew"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="title" local="title"
TEXT: nodeName="#text" nodeValue="XML Developer's Guide"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="genre" local="genre"
TEXT: nodeName="#text" nodeValue="Computer"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="price" local="price"
TEXT: nodeName="#text" nodeValue="44.95"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="publish_date" local="publish_date"
TEXT: nodeName="#text" nodeValue="2000-10-01"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="description" local="description"
TEXT: nodeName="#text" nodeValue="An in-depth look at creating applications
with XML.."
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="book" local="book"
ATTR: nodeName="id" local="id" nodeValue="bk102"
TEXT: nodeName="#text" nodeValue="bk102"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="author" local="author"
TEXT: nodeName="#text" nodeValue="Ralls, Kim"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="title" local="title"
TEXT: nodeName="#text" nodeValue="Midnight Rain"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="genre" local="genre"
TEXT: nodeName="#text" nodeValue="Fantasy"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="price" local="price"
TEXT: nodeName="#text" nodeValue="5.95"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="publish_date" local="publish_date"
TEXT: nodeName="#text" nodeValue="2000-12-16"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="description" local="description"
TEXT: nodeName="#text" nodeValue="A former architect battles corporate zombies,
```

Output

```
C:\Windows\system32\cmd.exe - run1.bat
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

D:\TESTAR>set CLASSPATH=%CLASSPATH%;.

D:\TESTAR>run1.bat

D:\TESTAR>set CLASSPATH=%CLASSPATH%;.;C:\TESTAR\orpc.dev\lib\orpc.jar

D:\TESTAR>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: books.xml 4550 bytes transferidos
Ficheiro gravado: books.xml
Closing socket.
```

File Server

C:\Windows\system32\cmd.exe

```
C:\TESTAR>java FileClient localhost books.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
books.xml 4550 bytes enviados
```

C:\TESTAR>

File Client

■ 2.1. Parser DOM

Integração de Sistemas

■ medsamp2014.xml

C:\Program Files\Common Files\Oracle\Java\javapath\java.exe

```
DOC: nodeName="#document"
ELEM: nodeName="MedlineCitationSet" local="MedlineCitationSet"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="MedlineCitation" local="MedlineCitation"
  ATTR: nodeName="Owner" local="Owner" nodeValue="NLM"
    TEXT: nodeName="#text" nodeValue="NLM"
  ATTR: nodeName="Status" local="Status" nodeValue="MEDLINE"
    TEXT: nodeName="#text" nodeValue="MEDLINE"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="PMID" local="PMID"
  ATTR: nodeName="Version" local="Version" nodeValue="1"
    TEXT: nodeName="#text" nodeValue="1"
  TEXT: nodeName="#text" nodeValue="10540283"
TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="DateCreated" local="DateCreated"
  TEXT: nodeName="#text" nodeValue=[WS]
  ELEM: nodeName="Year" local="Year"
    TEXT: nodeName="#text" nodeValue="1999"
  TEXT: nodeName="#text" nodeValue=[WS]
  ELEM: nodeName="Month" local="Month"
    TEXT: nodeName="#text" nodeValue="12"
  TEXT: nodeName="#text" nodeValue=[WS]
  ELEM: nodeName="Day" local="Day"
    TEXT: nodeName="#text" nodeValue="17"
  TEXT: nodeName="#text" nodeValue=[WS]
  TEXT: nodeName="#text" nodeValue=[WS]
ELEM: nodeName="DateCompleted" local="DateCompleted"
  TEXT: nodeName="#text" nodeValue=[WS]
  ELEM: nodeName="Year" local="Year"
    TEXT: nodeName="#text" nodeValue="1999"
  TEXT: nodeName="#text" nodeValue=[WS]
  ELEM: nodeName="Month" local="Month"
```

Output

Seletor Administrador: Linha de comandos - run1.bat

```
C:\TESTAR>run1.bat
C:\TESTAR>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.jar
C:\TESTAR>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.jar;.
C:\TESTAR>java FileServer
Netbul JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: medsamp2014.xml 61450 bytes transferidos
Ficheiro gravado: medsamp2014.xml
Closing socket.
```

File Server

Administrator: Linha de comandos

```
C:\TESTAR>run2.bat
C:\TESTAR>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.jar
C:\TESTAR>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.jar;.
C:\TESTAR>java FileClient localhost medsamp2014.xml
Netbul JavaRPC demo, not distributable!
Ligado ao servidor localhost
medsamp2014.xml 61450 bytes enviados
```

File Client

■ 2.2. Parser SAX

Integração de Sistemas

■ books.xml

Administrator: Linha de comandos

```
C:\TESTAR_SAX\sax>java SAXLocalNameCount.java "..\books.xml"
Note: SAXLocalNameCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: SAXLocalNameCount.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Local Name "genre" occurs 12 times
Local Name "catalog" occurs 1 times
Local Name "book" occurs 12 times
Local Name "author" occurs 12 times
Local Name "description" occurs 12 times
Local Name "price" occurs 12 times
Local Name "publish_date" occurs 12 times
Local Name "title" occurs 12 times
C:\TESTAR_SAX\sax>
```

Output

```
C:\Windows\system32\cmd.exe - run1.bat
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\TESTAR_SAX>run1.bat

C:\TESTAR_SAX>set CLASSPATH=C:\TESTAR_SAX\orpc.dev\lib\orpc.jar

C:\TESTAR_SAX>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: books.xml 4550 bytes transferidos
Ficheiro gravado: books.xml
Closing socket.
```

File Server


```
C:\Windows\system32\cmd.exe
C:\TESTAR_SAX>java FileClient localhost books.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
books.xml 4550 bytes enviados

C:\TESTAR_SAX>
```

File Client

■ 2.2. Parser SAX

Integração de Sistemas

■ note.xml

Administrator: Linha de comandos

```
C:\TESTAR_SAX>java FileClient localhost note.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
note.xml 145 bytes enviados

C:\TESTAR_SAX>cd sax

C:\TESTAR_SAX\sax>java SAXLocalNameCount.java "..\note.xml"
Note: SAXLocalNameCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: SAXLocalNameCount.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Local Name "body" occurs 1 times
Local Name "note" occurs 1 times
Local Name "heading" occurs 1 times
Local Name "from" occurs 1 times
Local Name "to" occurs 1 times

C:\TESTAR_SAX\sax>
```

Output

Administrator: Linha de comandos - run1.bat

```
C:\TESTAR_SAX>run1.bat

C:\TESTAR_SAX>set CLASSPATH=%CLASSPATH%;;C:\TESTAR_SAX\orpc.jar;C:\TESTAR_SAX\orpc.dev\lib\orpc.jar

C:\TESTAR_SAX>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: note.xml 145 bytes transferidos
Ficheiro gravado: note.xml
Closing socket.
```

File Server

```
C:\TESTAR_SAX>java FileClient localhost note.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
note.xml 145 bytes enviados
```

File Client

■ 2.2. Parser SAX

Integração de Sistemas

■ medsamp2013.xml

```
C:\TESTAR_SAX\sax>java SAXLocalNameCount.java ..\medsamp2013.xml
Note: SAXLocalNameCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: SAXLocalNameCount.java uses unchecked or unsafe operations. C:\TESTAR>run1.bat
Note: Recompile with -Xlint:unchecked for details.

Local Name "RefSource" occurs 9 times
Local Name "Grant" occurs 6 times
Local Name "NameOfSubstance" occurs 47 times
Local Name "AbstractText" occurs 18 times
Local Name "AccessionNumber" occurs 4 times
Local Name "AuthorList" occurs 11 times
Local Name "MeshHeadingList" occurs 10 times
Local Name "ForeName" occurs 328 times
Local Name "ArticleTitle" occurs 11 times
Local Name "MedlineCitationSet" occurs 1 times
Local Name "Volume" occurs 10 times
Local Name "ChemicalList" occurs 7 times
Local Name "Day" occurs 36 times
Local Name "Affiliation" occurs 11 times
Local Name "ISOAbbreviation" occurs 11 times
Local Name "Note" occurs 1 times
Local Name "Acronym" occurs 5 times
Local Name "CopyrightInformation" occurs 2 times
Local Name "DescriptorName" occurs 151 times
Local Name "PublicationType" occurs 25 times
Local Name "MedlineJournalInfo" occurs 11 times
Local Name "MedlineTA" occurs 11 times
Local Name "DateCompleted" occurs 10 times
Local Name "OtherID" occurs 2 times
Local Name "Title" occurs 11 times
Local Name "PMID" occurs 18 times
Local Name "MeshHeading" occurs 151 times
Local Name "CommentsCorrectionsList" occurs 3 times
```

Output

```
Administrator: C:\WINDOWS\system32\cmd.exe - run1.bat
C:\TESTAR>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.jar;C:\TE
pc.jar
C:\TESTAR>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: medsamp2014.xml 61440 bytes transferidos
Ficheiro gravado: medsamp2014.xml
Closing socket.
```

File Server

```
Administrator: C:\WINDOWS\system32\cmd.exe
C:\TESTAR>run2.bat
C:\TESTAR>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.ja
C:\TESTAR>java FileClient localhost medsamp2014.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
medsamp2014.xml 61440 bytes enviados
C:\TESTAR>
```

File Client

■ 2.3. Parser JDOM

Integração de Sistemas

books.xml

```
C:\Program Files\Common Files\Oracle\Java\javapath\java.exe
```

```
<-----XML DOCUMENT----->
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
    with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies,
    an evil sorceress, and her own childhood to become queen
    of the world.</description>
  </book>
  <book id="bk103">
    <author>Corets, Eva</author>
    <title>Maeve Ascendant</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-11-17</publish_date>
    <description>After the collapse of a nanotechnology
    society in England, the young survivors lay the
    foundation for a new society.</description>
  </book>
```

Output

```
Administrator: C:\WINDOWS\system32\cmd.exe - run1.bat
```

```
C:\TESTAR_JDOM>run1.bat
```

```
C:\TESTAR_JDOM>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.jar
C:\TESTAR_JDOM>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: books.xml 4548 bytes transferidos
Ficheiro gravado: books.xml
Closing socket.
```

File Server

```
Administrator: C:\WINDOWS\system32\cmd.exe
```

```
C:\TESTAR_JDOM>set CLASSPATH=%CLASSPATH%;C:\TESTAR\orpc.dev\lib\orpc.jar
C:\TESTAR_JDOM>java FileClient localhost books.xml
Error: Could not find or load main class FileClient
Caused by: java.lang.ClassNotFoundException: FileClient
```

```
C:\TESTAR_JDOM>set CLASSPATH=%CLASSPATH%;
```

```
C:\TESTAR_JDOM>java FileClient localhost books.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
books.xml 4548 bytes enviados
```

File Client

■ 2.3. Parser JDOM

Integração de Sistemas

■ note.xml

```
Selecionar C:\Windows\system32\cmd.exe - java Pla
-----XML DOCUMENT-----
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited by XMLSpy --><note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Output

```
C:\Windows\system32\cmd.exe - run1.bat
C:\TESTAR_JDOM>run1.bat
C:\TESTAR_JDOM>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\
c.jar
C:\TESTAR_JDOM>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: note.xml 145 bytes transferidos
Ficheiro gravado: note.xml
Closing socket.
```

File Server

```
C:\Windows\system32\cmd.exe
C:\TESTAR_JDOM>java FileClient localhost note.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
note.xml 145 bytes enviados
```

File Client

■ 2.3. Parser JDOM

Integração de Sistemas

■ medsamp2013.xml

```
C:\Windows\system32\cmd.exe -java PlayWithJDom medsamp2013.xml
<----- XML DOCUMENT ----->
<?xml version="1.0" encoding="UTF-8"?>
<MedlineCitationSet>
<MedlineCitation Owner="NLM" Status="MEDLINE">
<PMID Version="1">10540283</PMID>
<DateCreated>
<Year>1999</Year>
<Month>12</Month>
<Day>17</Day>
</DateCreated>
<DateCompleted>
<Year>1999</Year>
<Month>12</Month>
<Day>17</Day>
</DateCompleted>
<DateRevised>
<Year>2006</Year>
<Month>11</Month>
<Day>15</Day>
</DateRevised>
<Article PubModel="Print">
<Journal>
<ISSN IssnType="Print">0950-382X</ISSN>
<JournalIssue CitedMedium="Print">
<Volume>34</Volume>
<Issue>1</Issue>
<PubDate>
<Year>1999</Year>
<Month>Oct</Month>
</PubDate>
</JournalIssue>
<Title>Molecular microbiology</Title>
<ISOAbbreviation>Mol. Microbiol.</ISOAbbreviation>
</Journal>
<ArticleTitle>Transcription regulation of the nir gene cluster encoding nitrite
<Pagination>
<MedlinePgn>24-36</MedlinePgn>
</Pagination>
```

Output

```
C:\Windows\system32\cmd.exe - run1.bat
C:\TESTAR_JDOM>run1.bat
C:\TESTAR_JDOM>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.jar;C:\TESTA
c.jar;C:\TESTAR\orpc.dev\lib\orpc.jar
C:\TESTAR_JDOM>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: medsamp2013.xml 77824 bytes transferidos
Ficheiro gravado: medsamp2013.xml
Closing socket.
```

File Server

```
C:\Windows\system32\cmd.exe
C:\TESTAR_JDOM>java FileClient localhost medsamp2013.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
medsamp2013.xml 77824 bytes enviados
```

C:\TESTAR_JDOM>

File Client

■ 2.4. Parser JAXP

Integração de Sistemas

books.xml

C:\Program Files\Common Files\Oracle\Java\javapath\java.exe

```
FACTORY: com.sun.xml.internal.stream.XMLInputFactoryImpl@1324409e
filename = books.xml

<?xml version="1.0" encoding="null"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
      with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies,
      an evil sorceress, and her own childhood to become queen
      of the world.</description>
  </book>
  <book id="bk103">
    <author>Corets, Eva</author>
    <title>Maeve Ascendant</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-11-17</publish_date>
    <description>After the collapse of a nanotechnology
      society in England, the young survivors lay the
      foundation for a new society.</description>
  </book>
```

Output

Administrator: C:\WINDOWS\system32\cmd.exe - run1.bat

C:\TESTAR_JAXP>run1.bat

C:\TESTAR_JAXP>set CLASSPATH=C:\TESTAR\orpc.dev\lib\orpc.jar

C:\TESTAR_JAXP>java FileServer
 Netbula JavaRPC demo, not distributable!
 Registered TCP transport.
 Registered UDP transport.
 Ficheiro recebido: books.xml 4550 bytes transferidos
 Ficheiro gravado: books.xml
 Closing socket.

File Server

C:\Windows\system32\cmd.exe

C:\TESTAR_JAXP>java FileClient localhost books.xml
 Netbula JavaRPC demo, not distributable!
 Ligado ao servidor localhost
 books.xml 4550 bytes enviados

C:\TESTAR_JAXP>

File Client

■ 2.4. Parser JAXP

Integração de Sistemas

■ simple.xml

C:\Program Files\Common Files\Oracle\Java\javapath\java.exe

```
FACTORY: com.sun.xml.internal.stream.XMLInputFactoryImpl@1324409e
filename = simple.xml

<?xml version="1.0" encoding="ISO-8859-1"?>
Edited by XMLSpy? <!-- Edited by XMLSpy? --><breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
    <calories>650</calories>
  </food>
  <food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>Light Belgian waffles covered with strawberries and whipped cream</description>
    <calories>900</calories>
  </food>
  <food>
    <name>Berry-Berry Belgian Waffles</name>
    <price>$8.95</price>
    <description>Light Belgian waffles covered with an assortment of fresh berries and whipped cream</description>
    <calories>900</calories>
  </food>
  <food>
    <name>French Toast</name>
    <price>$4.50</price>
    <description>Thick slices made from our homemade sourdough bread</description>
    <calories>600</calories>
  </food>
  <food>
    <name>Homestyle Breakfast</name>
    <price>$6.95</price>
    <description>Two eggs, bacon or sausage, toast, and our ever-popular hash browns</description>
    <calories>950</calories>
  </food>
</breakfast_menu> Parsing Time = 29
```

Output

C:\ Administrator: C:\WINDOWS\system32\cmd.exe - run1.bat

C:\TESTAR_JAXP>run1.bat

C:\TESTAR_JAXP>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpcorpc.jar

C:\TESTAR_JAXP>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: simple.xml 1135 bytes transferidos
Ficheiro gravado: simple.xml
Closing socket.

File Server

C:\Windows\system32\cmd.exe

C:\TESTAR_JAXP>java FileClient localhost simple.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
simple.xml 1135 bytes enviados

C:\TESTAR_JAXP>

File Client

■ 2.4. Parser JAXP

Integração de Sistemas

■ medsamp2014.xml

C:\Program Files\Common Files\Oracle\Java\javapath\java.exe

```
FACTORY: com.sun.xml.internal.stream.XMLInputFactoryImpl@1324409e
filename = medsamp2014.xml

<?xml version="1.0" encoding="UTF-8"?>
<MedlineCitationSet>
<MedlineCitation Owner="NLM" Status="MEDLINE">
<PMID Version="1">10540283</PMID>
<DateCreated>
<Year>1999</Year>
<Month>12</Month>
<Day>17</Day>
</DateCreated>
<DateCompleted>
<Year>1999</Year>
<Month>12</Month>
<Day>17</Day>
</DateCompleted>
<DateRevised>
<Year>2006</Year>
<Month>11</Month>
<Day>15</Day>
</DateRevised>
<Article PubModel="Print">
<Journal>
<ISSN IssnType="Print">0950-382X</ISSN>
<JournalIssue CitedMedium="Print">
<Volume>34</Volume>
<Issue>1</Issue>
<PubDate>
<Year>1999</Year>
<Month>Oct</Month>
</PubDate>
</JournalIssue>
<Title>Molecular microbiology</Title>
<ISOAbbreviation>Mol. Microbiol.</ISOAbbreviation>
</Journal>
<ArticleTitle>Transcription regulation of the nir gene cluster encoding nitrite reductase</ArticleTitle>
<Pagination>
```

Administrator: C:\WINDOWS\system32\cmd.exe - run1.bat

C:\TESTAR_JAXP>run1.bat

C:\TESTAR_JAXP>set CLASSPATH=;C:\TESTAR\orpc.dev\lib\orpc.jar;C:\TESTAR\orpc.dev\lib\orpc.jar;C:\TESTAR\orpc.dev\lib\orpc.jar

C:\TESTAR_JAXP>java FileServer
Netbula JavaRPC demo, not distributable!
Registered TCP transport.
Registered UDP transport.
Ficheiro recebido: medsamp2014.xml 102321 bytes transferidos
Ficheiro gravado: medsamp2014.xml
Closing socket.

File Server

C:\Windows\system32\cmd.exe

C:\TESTAR_JAXP>java FileClient localhost medsamp2014.xml
Netbula JavaRPC demo, not distributable!
Ligado ao servidor localhost
medsamp2014.xml 102321 bytes enviados

File Client

Output

■ 2.5. Parser JAXB

Integração de Sistemas

C:\Program Files\Common Files\Oracle\Java\javapath\java.exe

Ship the following items to:

Alice Smith
123 Maple Street
Cambridge, MA 12345
US

5 copies of "Nosferatu - Special Edition (1929)"

3 copies of "The Mummy (1959)"

3 copies of "Godzilla and Mothra: Battle for Earth/Godzilla vs. King Ghidora"

C:\Windows\system32\cmd.exe - run1.bat

C:\TESTAR_JAXB\samples\unmarshal-read>run1.bat

C:\TESTAR_JAXB\samples\unmarshal-read>java FileServer
Netbula JavaRPC demo, not distributable!

Registered TCP transport.

Registered UDP transport.

Ficheiro recebido: po.xml 2835 bytes transferidos

Ficheiro gravado: po.xml

Closing socket.

Output

File Server

C:\Windows\system32\cmd.exe

C:\TESTAR_JAXB\samples\unmarshal-read>run2.bat

C:\TESTAR_JAXB\samples\unmarshal-read>java FileClient localhost po.xml
Netbula JavaRPC demo, not distributable!

Ligado ao servidor localhost

po.xml 2835 bytes enviados

File Client

■ 3. SOAP and REST WebServices Implementation integrating XML files in a Data Base

Integração de Sistemas

- In this section, it is described the second part of the 8th tutorial, which consists of testing and analyzing the given source code for different webservices created in JAVA and PHP, using some of the parsers tested in the first part of the tutorial (correspondent to the 7th tutorial).

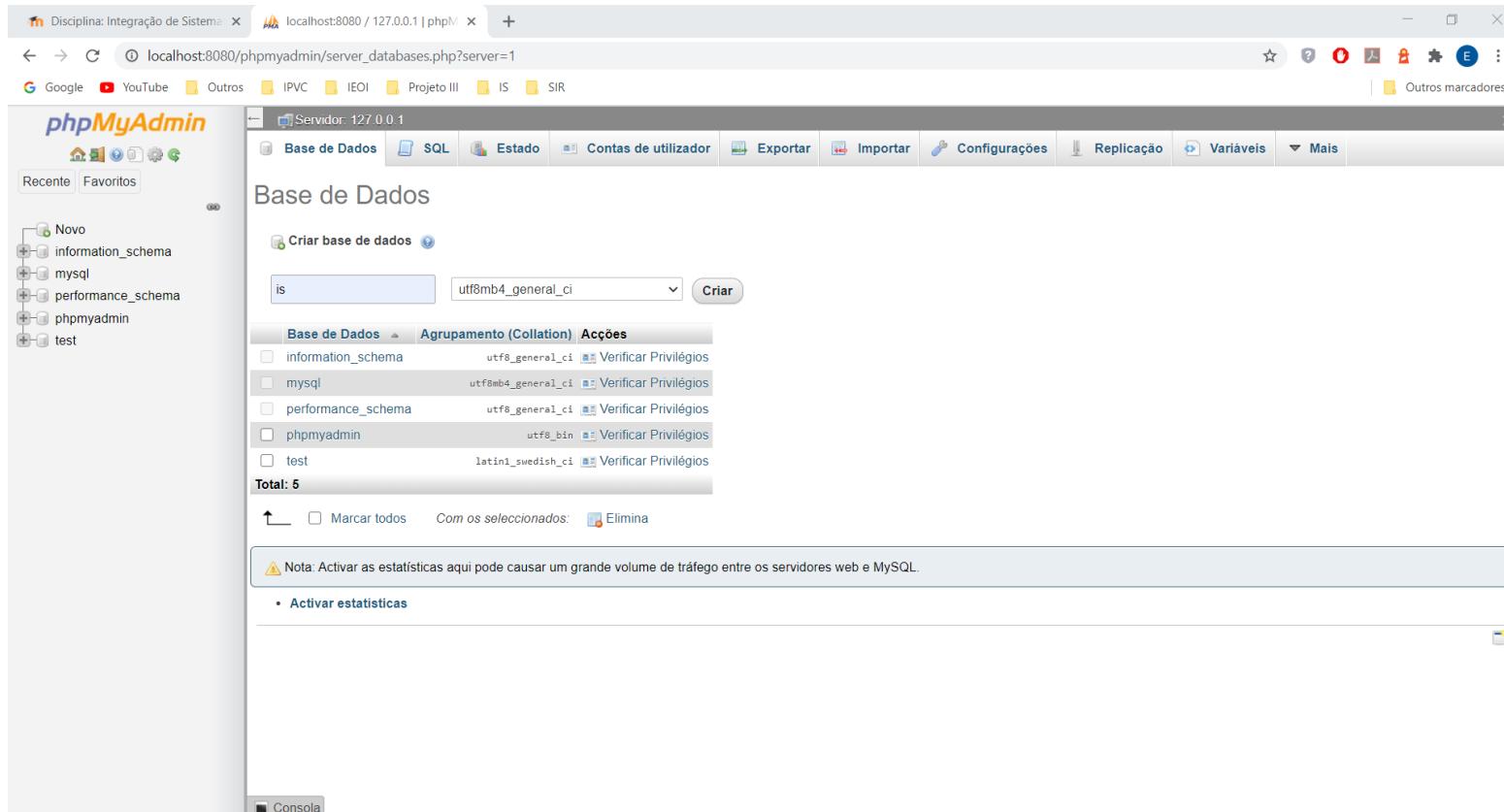
- In order to start this part of the tutorial, it was required to adjust some configurations to the NetBeans IDE to allow webservices to be tested locally in the already installed glassfish server, as well as creating a new MySQL database which runs on the local server provided by XAMPP, with one table called Alunos, and two columns, one for the id (which is the primary key), and another one for the student's name, this to match the xml file alunos.xml which will be used later in the tutorial.

■ 3. SOAP and REST WebServices Implementation

integrating XML files in a Data Base

Integração de Sistemas

■ Database creation using phpMyAdmin



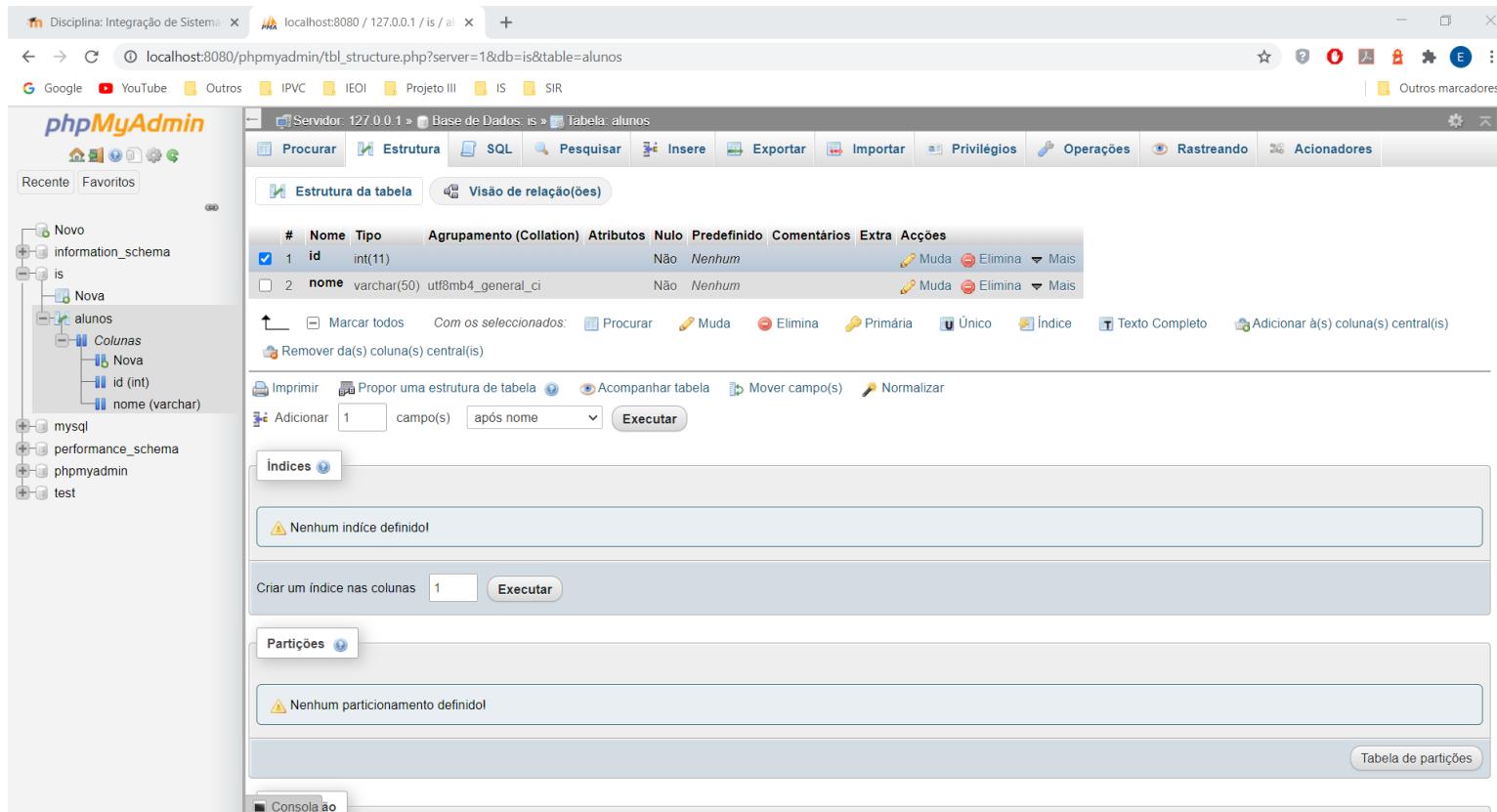
The screenshot shows the phpMyAdmin interface on a web browser. The URL is `localhost:8080/phpmyadmin/server_databases.php?server=1`. The main menu bar includes links for Google, YouTube, Outros, IPVC, IEOI, Projeto III, IS, and SIR. The left sidebar shows the database structure with 'Novo' selected. The central panel displays the 'Base de Dados' section, where a new database named 'is' is being created with 'utf8mb4_general_ci' collation. A note at the bottom states: 'Nota: Activar as estatísticas aqui pode causar um grande volume de tráfego entre os servidores web e MySQL.' (Note: Activating statistics here may cause a large volume of traffic between web servers and MySQL.)

■ 3. SOAP and REST WebServices Implementation

integrating XML files in a Data Base

Integração de Sistemas

■ Table “alunos” and columns “id” and “nome” created in the new database



The screenshot shows the phpMyAdmin interface for a database named 'is'. The left sidebar shows the database structure with a schema named 'Nova' containing a table 'alunos' with two columns: 'id' (int) and 'nome' (varchar). The main panel displays the 'Estrutura da tabela' (Table Structure) for the 'alunos' table. The columns are listed with their names, types, and various configuration options like collation, attributes, and nullability.

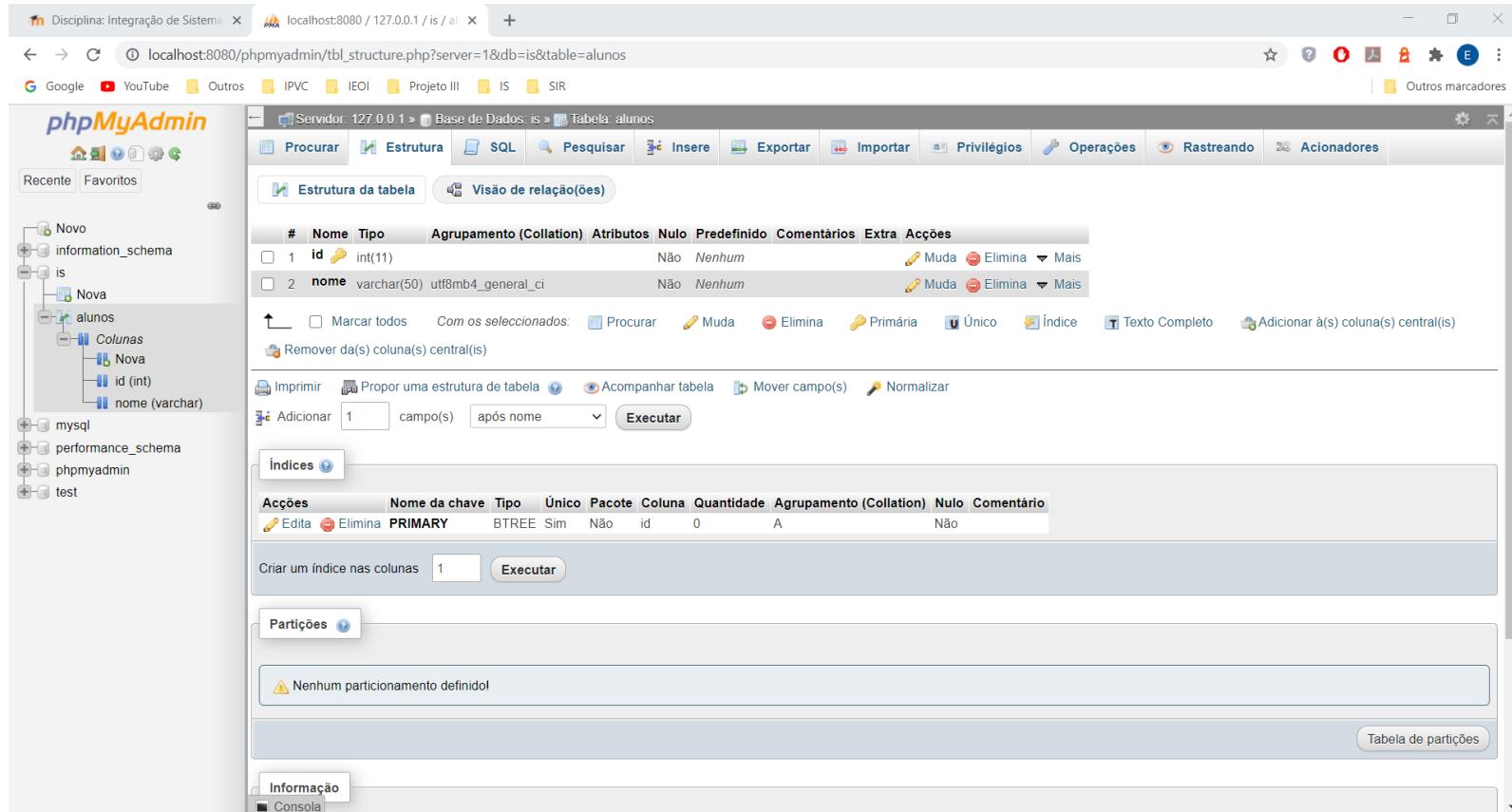
#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input checked="" type="checkbox"/>	1	id	int(11)		Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	2	nome	varchar(50)	utf8mb4_general_ci	Não	Nenhum			Muda Elimina Mais

■ 3. SOAP and REST WebServices Implementation

integrating XML files in a Data Base

Integração de Sistemas

■ Definition of the column “id” as the table’s primary key



The screenshot shows the phpMyAdmin interface for the 'alunos' table in the 'is' database. The 'Estrutura da tabela' (Table Structure) tab is selected. The table has two columns: 'id' (int(11)) and 'nome' (varchar(50)). The 'id' column is defined as the primary key (PRIMARY). The 'Índices' (Indexes) tab shows a single index named 'id' (PRIMARY, BTREE).

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
1	id	int(11)			Não	Nenhum			Muda Elimina Mais
2	nome	varchar(50)	utf8mb4_general_ci		Não	Nenhum			Muda Elimina Mais

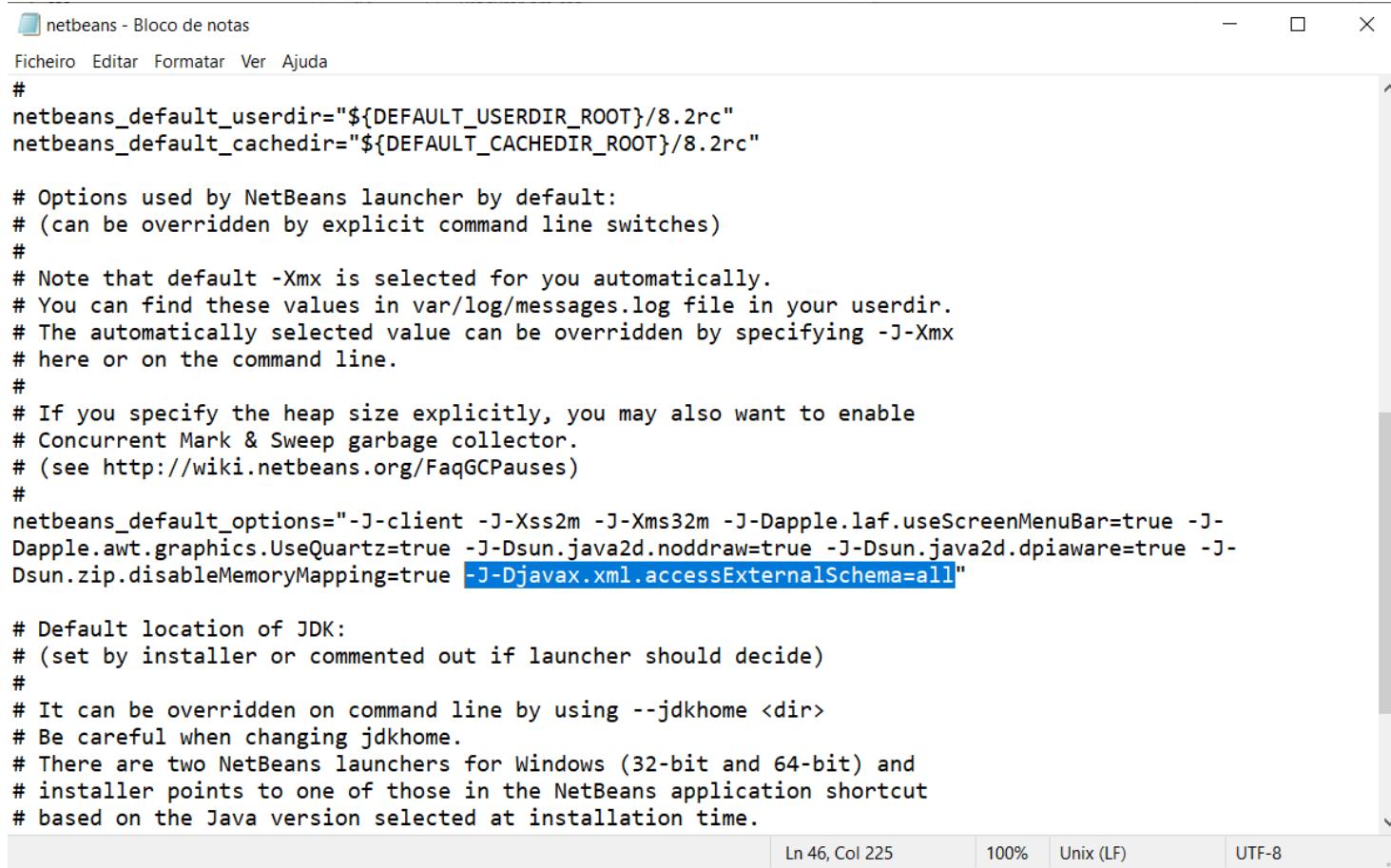
Acções	Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo	Comentário
Edita Elimina	PRIMARY	BTREE	Sim	Não	id	0	A	Não	

■ 3. SOAP and REST WebServices Implementation

integrating XML files in a Data Base

Integração de Sistemas

■ Option added to NetBeans default configuration



```

netbeans - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
#
netbeans_default_userdir="${DEFAULT_USERDIR_ROOT}/8.2rc"
netbeans_default_cachedir="${DEFAULT_CACHEDIR_ROOT}/8.2rc"

# Options used by NetBeans launcher by default:
# (can be overridden by explicit command line switches)
#
# Note that default -Xmx is selected for you automatically.
# You can find these values in var/log/messages.log file in your userdir.
# The automatically selected value can be overridden by specifying -J-Xmx
# here or on the command line.
#
# If you specify the heap size explicitly, you may also want to enable
# Concurrent Mark & Sweep garbage collector.
# (see http://wiki.netbeans.org/FaqGCPauses)
#
netbeans_default_options="-J-client -J-Xss2m -J-Xms32m -J-Dapple.laf.useScreenMenuBar=true -J-
Dapple.awt.graphics.UseQuartz=true -J-Dsun.java2d.nodraw=true -J-Dsun.java2d.dpiaware=true -J-
Dsun.zip.disableMemoryMapping=true -J-Djax.xml.accessExternalSchema=all

# Default location of JDK:
# (set by installer or commented out if launcher should decide)
#
# It can be overridden on command line by using --jdkhome <dir>
# Be careful when changing jdkhome.
# There are two NetBeans launchers for Windows (32-bit and 64-bit) and
# installer points to one of those in the NetBeans application shortcut
# based on the Java version selected at installation time.

```

Ln 46, Col 225 100% Unix (LF) UTF-8

■ 3.1. Java webservices implementation

Integração de Sistemas

■ This section demonstrates the output of the execution of SOAP and REST webservices using the NetBeans IDE, executing them in the glassfish server along with a MySQL database with the information present in the alunos.xml file. The result of the execution of the webservices is demonstrated by screenshots, as well as the headers of the request and the server's response, verified on the browser's developer tools.

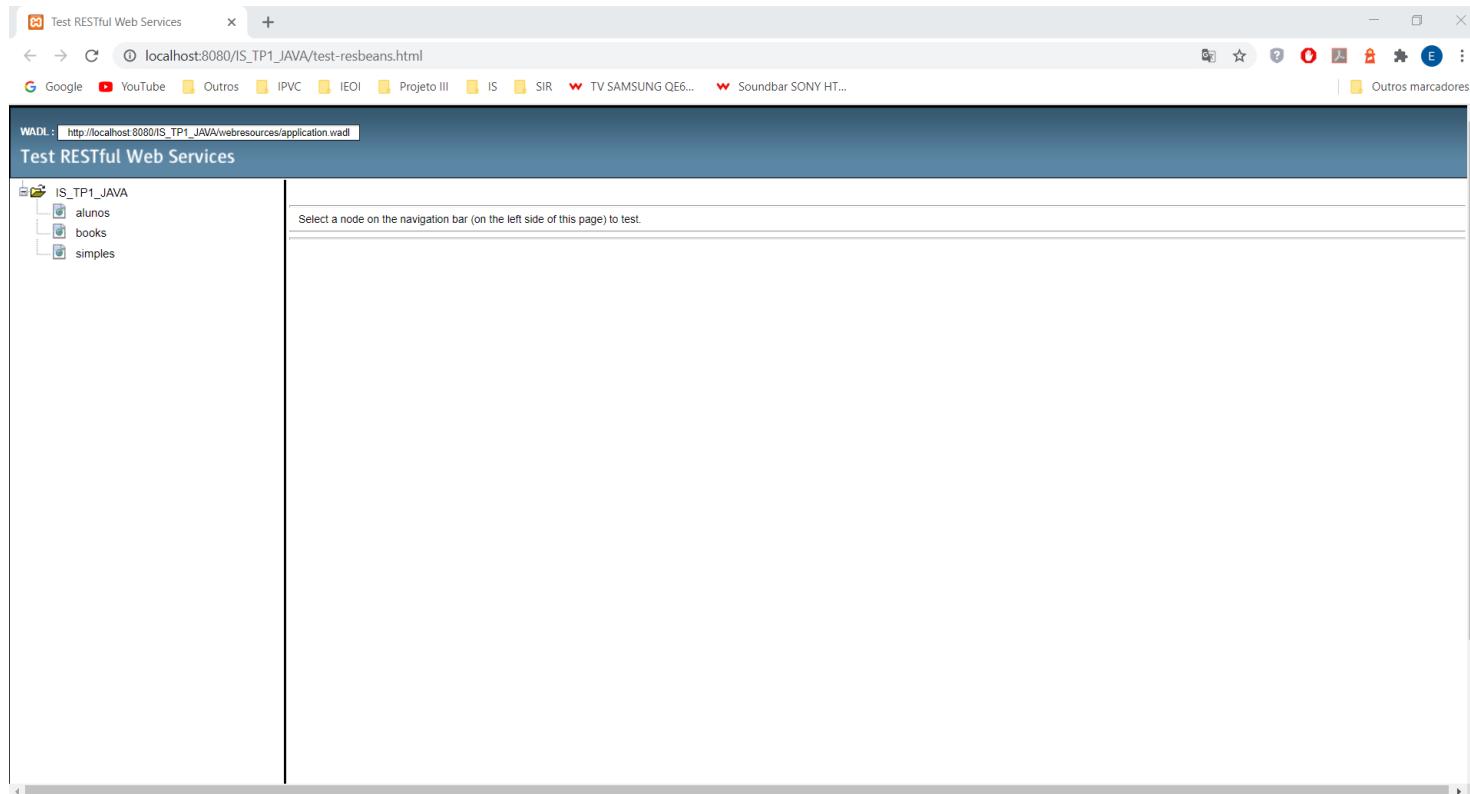
■ These webservices are implemented using three different Java classes:

- One for the declaration of the class that defines an object present in the XML file, with all the XML tags declared as attributes of this same class, as well as all the necessary getters and setters;
- The second one for the declaration of the array of objects defined in the first class, being the match for the root tag in the XML file;
- The third corresponds to the definition of the webservice, implementing the method to get all the information from the XML file, using a parser to de-serialize its contents, and create java objects from the different XML tags.

■ 3.1. Java webservices implementation

Integração de Sistemas

■ Available REST webservices



The screenshot shows a web browser window titled "Test RESTful Web Services" with the URL "localhost:8080/IS_TP1_JAVA/test-resbeans.html". The page displays a navigation bar on the left with nodes: IS_TP1_JAVA (selected), alunos, books, and simples. A message in the center says "Select a node on the navigation bar (on the left side of this page) to test."

■ 3.1. Java webservices implementation

Integração de Sistemas

■ Available SOAP webservices

Endpoint	Information
Service Name: {http://is2.com/}Books_ws_soap Port Name: {http://is2.com/}Books_ws_soapPort	Address: http://localhost:8080/IS_TP1_JAVA/Books_ws_soap WSDL: http://localhost:8080/IS_TP1_JAVA/Books_ws_soap?wsdl Implementation class: com.is2.Books_ws_soap
Service Name: {http://is2.com/}Simples_ws_soap Port Name: {http://is2.com/}Simples_ws_soapPort	Address: http://localhost:8080/IS_TP1_JAVA/Simples_ws_soap WSDL: http://localhost:8080/IS_TP1_JAVA/Simples_ws_soap?wsdl Implementation class: com.is2.Simples_ws_soap
Service Name: {http://db.is.com/}Alunos_ws_soap Port Name: {http://db.is.com/}Alunos_ws_soapPort	Address: http://localhost:8080/IS_TP1_JAVA/Alunos_ws_soap WSDL: http://localhost:8080/IS_TP1_JAVA/Alunos_ws_soap?wsdl Implementation class: com.is.db.Alunos_ws_soap

■ 3.1.1. SOAP webservice with file books.xml and JAXB parser

Integração de Sistemas

Method invocation trace X +

localhost:8080/IS_TP1_JAVA/Books_ws_soap?Tester

Google YouTube Outros IPVC IEOI Projeto III IS SIR Outros marcadores

Method returned

```
java.util.List : "[com.is2.Book@7d407828, com.is2.Book@236f6df4, com.is2.Book@1559586c, com.is2.Book@744f946, com.is2.Book@79bb40be, com.is2.Book@216382b0, com.is2.Book@7e2634c8, com.is2.Book@5e887c83, com.is2.Book@7d95795c, com.is2.Book@405e125c, com.is2.Book@f91eeeb, com.is2.Book@12e0e7fe]"
```

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
    <ns2:getBooks xmlns:ns2="http://is2.com/" />
</S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
    <ns2:getBooksResponse xmlns:ns2="http://is2.com/">
        <return>
            <author>Gambardella, Matthew</author>
            <title>XML Developer's Guide</title>
            <genre>Computer</genre>
            <price>44.95</price>
            <publish_date>2000-10-01</publish_date>
            <description>An in-depth look at creating applications with XML.</description>
        </return>
        <return>
            <author>Ralls, Kim</author>
            <title>Midnight Rain</title>
            <genre>Fantasy</genre>
            <price>5.95</price>
            <publish_date>2000-12-16</publish_date>
            <description>A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.</description>
        </return>
        <return>
```

■ 3.1.1. SOAP webservice with file books.xml and JAXB parser

Integração de Sistemas

Two screenshots of the Network tab in the Chrome DevTools developer console are shown side-by-side, illustrating the interaction between a client and a SOAP web service.

Left Screenshot (Client Request):

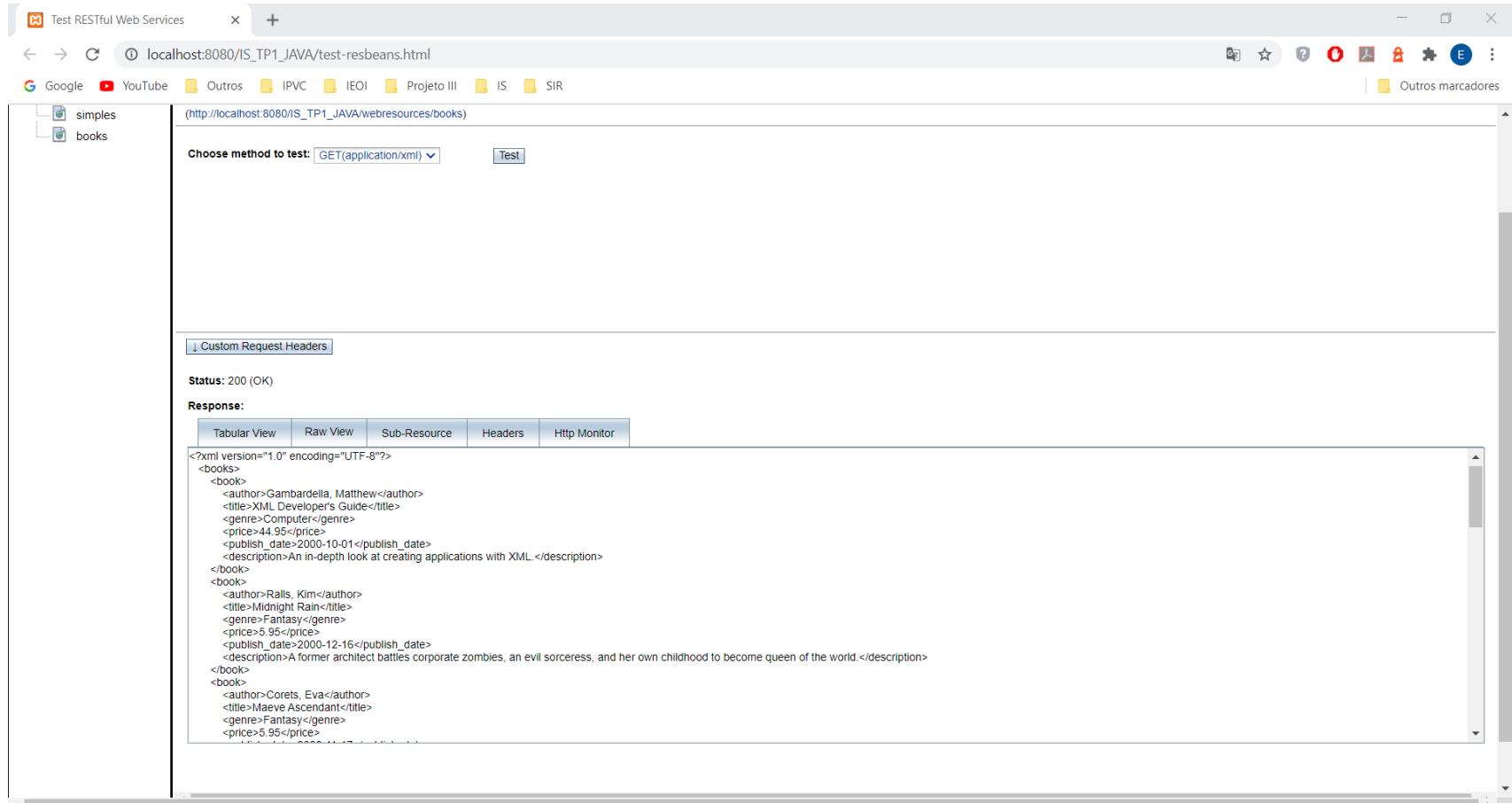
- Timing:** Shows a total duration of approximately 200 ms, with segments at 50 ms, 100 ms, 150 ms, and 200 ms.
- Name:** Headers section shows the following request headers:
 - Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
 - Cache-Control: max-age=0
 - Connection: keep-alive
 - Content-Length: 15
 - Content-Type: application/x-www-form-urlencoded
 - Host: localhost:8080
 - Origin: http://localhost:8080
 - Referer: http://localhost:8080/IS_TP1_JAVA/Books_ws_soap?Tester
 - sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome";v="86"
 - sec-ch-ua-mobile: ?0
 - Sec-Fetch-Dest: document
 - Sec-Fetch-Mode: navigate
 - Sec-Fetch-Site: same-origin
 - Sec-Fetch-User: ?1
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36
- Query String Parameters:** view source, view URL encoded. Tester: (empty)
- Form Data:** view source, view URL encoded. action: getBooks

Right Screenshot (Service Response):

- Timing:** Shows a total duration of approximately 200 ms, with segments at 50 ms, 100 ms, 150 ms, and 200 ms.
- Name:** Headers section shows the following response headers:
 - 1 <HTML lang=pt><HEAD><TITLE>Method invocation trace</TITLE></HEAD><H2>getBooks</H2><HR>
 - 2 <SOAP-ENV:Header>
 - 3 <S:Body>
 - 4 <ns2:getBooks xmlns:ns2="http://is2.com/">
 - 5 </S:Body>
 - 6 </S:Envelope>
 - 7 </pre></blockquote><HR><h4>SOAP Response</h4><HR><blockquote><pre xml:lang=<#>pt;>
 - 8 <SOAP-ENV:Header>
 - 9 <S:Body>
 - 10 <ns2:getBooksResponse xmlns:ns2="http://is2.com/">
 - 11 </S:Body>
 - 12 </S:Envelope>
 - 13 </pre></blockquote><HR></HTML>
- Text:** Line 13, Column 31

■ 3.1.2. REST webservice with file books.xml and JAXB parser

Integração de Sistemas



The screenshot shows a web browser window titled "Test RESTful Web Services". The address bar contains "localhost:8080/IS_TP1_JAVA/test-resbeans.html". The page content is a REST API test tool interface.

Request Configuration:

- Method: GET(application/xml)
- URL: (http://localhost:8080/IS_TP1_JAVA/webresources/books)

Status: 200 (OK)

Response:

Tabular View Raw View Sub-Resource Headers Http Monitor

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
  <book>
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications with XML.</description>
  </book>
  <book>
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.</description>
  </book>
  <book>
    <author>Corets, Eva</author>
    <title>Maeve Ascendant</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
  </book>
</books>
```

■ 3.1.3. SOAP webservice with file simple.xml and JAXB parser

Integração de Sistemas

```

Method invocation trace   +
localhost:8080/IS_TP1_JAVA/Simples_ws_soap?Tester
Google YouTube Outros IPVC IEOI Projeto III IS SIR Outros marcadores

Method returned
java.util.List : "[com.is2.Simple@3a5048c6, com.is2.Simple@6d5f1df3, com.is2.Simple@1dce9b69, com.is2.Simple@318e6db5, com.is2.Simple@9a0053a]"

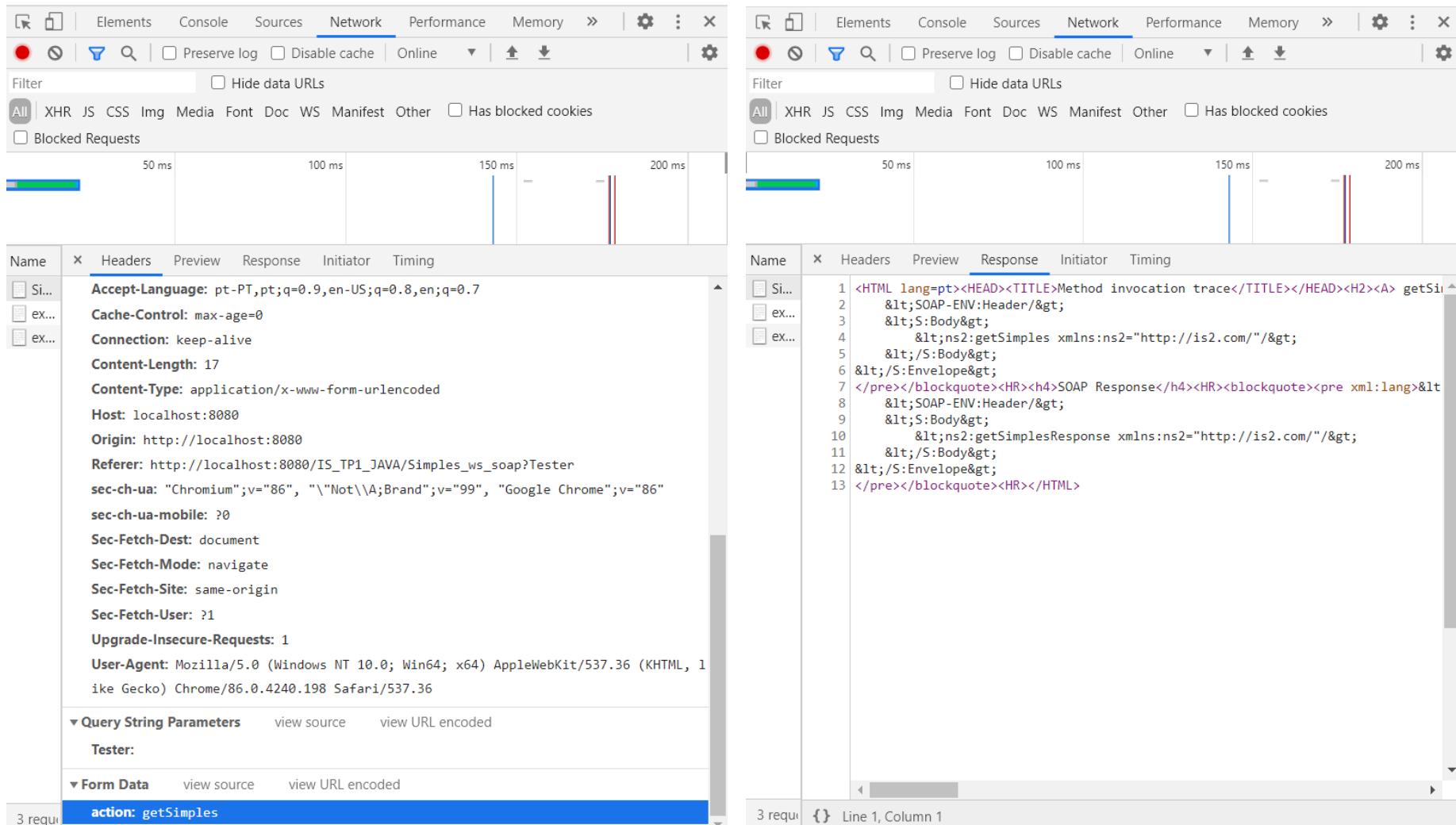
SOAP Request
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:getSimples xmlns:ns2="http://is2.com/">
</S:Body>
</S:Envelope>

SOAP Response
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:getSimplesResponse xmlns:ns2="http://is2.com/">
<return>
<name>Belgian Waffles</name>
<price>0.0</price>
<description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
<calories>650</calories>
</return>
<return>
<name>Strawberry Belgian Waffles</name>
<price>0.0</price>
<description>Light Belgian waffles covered with strawberries and whipped cream</description>
<calories>900</calories>
</return>
<return>
<name>Berry-Berry Belgian Waffles</name>
<price>0.0</price>
<description>Light Belgian waffles covered with an assortment of fresh berries and whipped cream</description>
<calories>900</calories>
</return>
<return>
<name>French Toast</name>
<price>0.0</price>

```

■ 3.1.3. SOAP webservice with file simple.xml and JAXB parser

Integração de Sistemas



The image shows two side-by-side browser developer tool windows, specifically focusing on the Network tab. Both windows have a similar layout with tabs for Elements, Console, Sources, Network (which is selected), Performance, Memory, and other developer tools.

Left Window (Request):

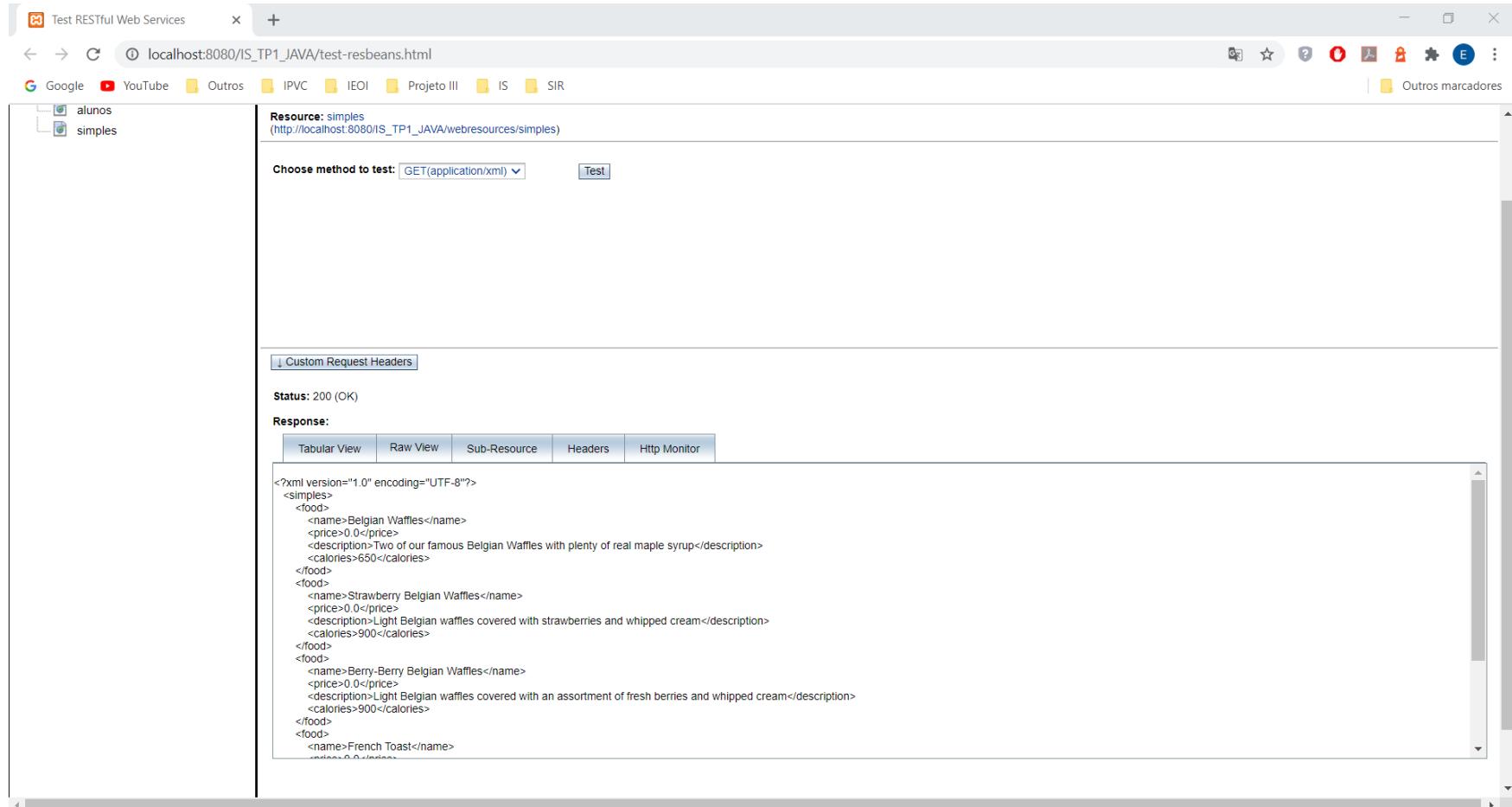
- Timing:** Shows a timeline with four main segments: 50 ms, 100 ms, 150 ms, and 200 ms. The first segment has a blue bar at the bottom. The second segment has a green bar at the bottom. The third segment has a red bar at the bottom. The fourth segment has a blue bar at the bottom.
- Name:** A table header with columns: Name, Headers, Preview, Response, Initiator, Timing.
- Headers:** A list of request headers:
 - Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
 - Cache-Control: max-age=0
 - Connection: keep-alive
 - Content-Length: 17
 - Content-Type: application/x-www-form-urlencoded
 - Host: localhost:8080
 - Origin: http://localhost:8080
 - Referer: http://localhost:8080/IS_TP1_JAVA/Simples_ws_soap?Tester
 - sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome";v="86"
 - sec-ch-ua-mobile: ?0
 - Sec-Fetch-Dest: document
 - Sec-Fetch-Mode: navigate
 - Sec-Fetch-Site: same-origin
 - Sec-Fetch-User: ?1
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36
- Query String Parameters:** view source, view URL encoded. Tester: (empty).
- Form Data:** view source, view URL encoded. action: getSimple.

Right Window (Response):

- Timing:** Shows a timeline with four main segments: 50 ms, 100 ms, 150 ms, and 200 ms. The first segment has a blue bar at the bottom. The second segment has a green bar at the bottom. The third segment has a red bar at the bottom. The fourth segment has a blue bar at the bottom.
- Name:** A table header with columns: Name, Headers, Preview, Response, Initiator, Timing.
- Headers:** A list of response headers:
 - 1 <HTML lang=pt><HEAD><TITLE>Method invocation trace</TITLE></HEAD><H2>getSimple
 - 2 <SOAP-ENV:Header>
 - 3 <S:Body>
 - 4 <ns2:getSimple xmlns:ns2="http://is2.com/">
 - 5 </S:Body>
 - 6 </S:Envelope>
 - 7 </pre></blockquote><HR><h4>SOAP Response</h4><HR><blockquote><pre xml:lang="en-US"><ns2:response xmlns:ns2="http://is2.com/">
 - 8 <SOAP-ENV:Header>
 - 9 <S:Body>
 - 10 <ns2:getSimpleResponse xmlns:ns2="http://is2.com/">
 - 11 </S:Body>
 - 12 </S:Envelope>
 - 13 </pre></blockquote><HR></HTML>
- Preview:** Shows the raw XML response content.
- Response:** Shows the raw XML response content.
- Initiator:** Shows the raw XML response content.
- Timing:** Shows the raw XML response content.

■ 3.1.4. REST webservice with file simple.xml and JAXB parser

Integração de Sistemas



Resource: simples
(http://localhost:8080/IS_TP1_JAVA/webservices/simple)

Choose method to test:

Status: 200 (OK)

Response:

Tabular View	Raw View	Sub-Resource	Headers	Http Monitor
--------------	----------	--------------	---------	--------------

```
<?xml version="1.0" encoding="UTF-8"?>
<simples>
<food>
<name>Belgian Waffles</name>
<price>0.0</price>
<description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
<calories>650</calories>
</food>
<food>
<name>Strawberry Belgian Waffles</name>
<price>0.0</price>
<description>Light Belgian waffles covered with strawberries and whipped cream</description>
<calories>900</calories>
</food>
<food>
<name>Berry-Berry Belgian Waffles</name>
<price>0.0</price>
<description>Light Belgian waffles covered with an assortment of fresh berries and whipped cream</description>
<calories>900</calories>
</food>
<food>
<name>French Toast</name>
<price>0.0</price>
<description>French toast made with our own bread, topped with syrup and whipped cream</description>
<calories>900</calories>
</food>
```

■ 3.1.5. SOAP webservice with file alunos.xml from database

Integração de Sistemas

Method invocation trace +

localhost:8080/IS_TP1_JAVA/Alunos_ws_soap?Tester

Method returned

```
java.util.List : "[com.is.db.Alunos@1d0ab8b8, com.is.db.Alunos@5d95492c, com.is.db.Alunos@4ecae9a9, com.is.db.Alunos@121365e8]"
```

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:getAlunos xmlns:ns2="http://db.is.com/">
</S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:getAlunosResponse xmlns:ns2="http://db.is.com/">
<return>
<id>1</id>
<nome>Antonio Silva</nome>
</return>
<return>
<id>2</id>
<nome>Maria Albertina</nome>
</return>
<return>
<id>3</id>
<nome>Jose Ferreira</nome>
</return>
<return>
<id>4</id>
<nome>Pedro Costa</nome>
</return>
</ns2:getAlunosResponse>
</S:Body>
</S:Envelope>
```

■ 3.1.5. SOAP webservice with file alunos.xml from database

Integração de Sistemas

Two screenshots of the Network tab in the Chrome DevTools developer console are shown side-by-side, illustrating the interaction with a SOAP web service.

Left Screenshot:

- Network Tab:** Shows a timeline with several requests. One request is highlighted in blue, showing a duration of approximately 20 ms.
- Request Headers:**

```

Accept-Encoding: gzip, deflate, br
Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
Cache-Control: max-age=0
Connection: keep-alive
Content-Length: 16
Content-Type: application/x-www-form-urlencoded
Host: localhost:8080
Origin: http://localhost:8080
Referer: http://localhost:8080/IS_TP1_JAVA/Alunos_ws_soap?Tester
sec-ch-ua: "Chromium";v="86", "\Not\\A;Brand";v="99", "Google Chrome";v="86"
sec-ch-ua-mobile: ?0
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36
  
```
- Form Data:**

action: getAlunos

Right Screenshot:

- Network Tab:** Shows a timeline with several requests. One request is highlighted in blue, showing a duration of approximately 20 ms.
- Response Headers:**

```

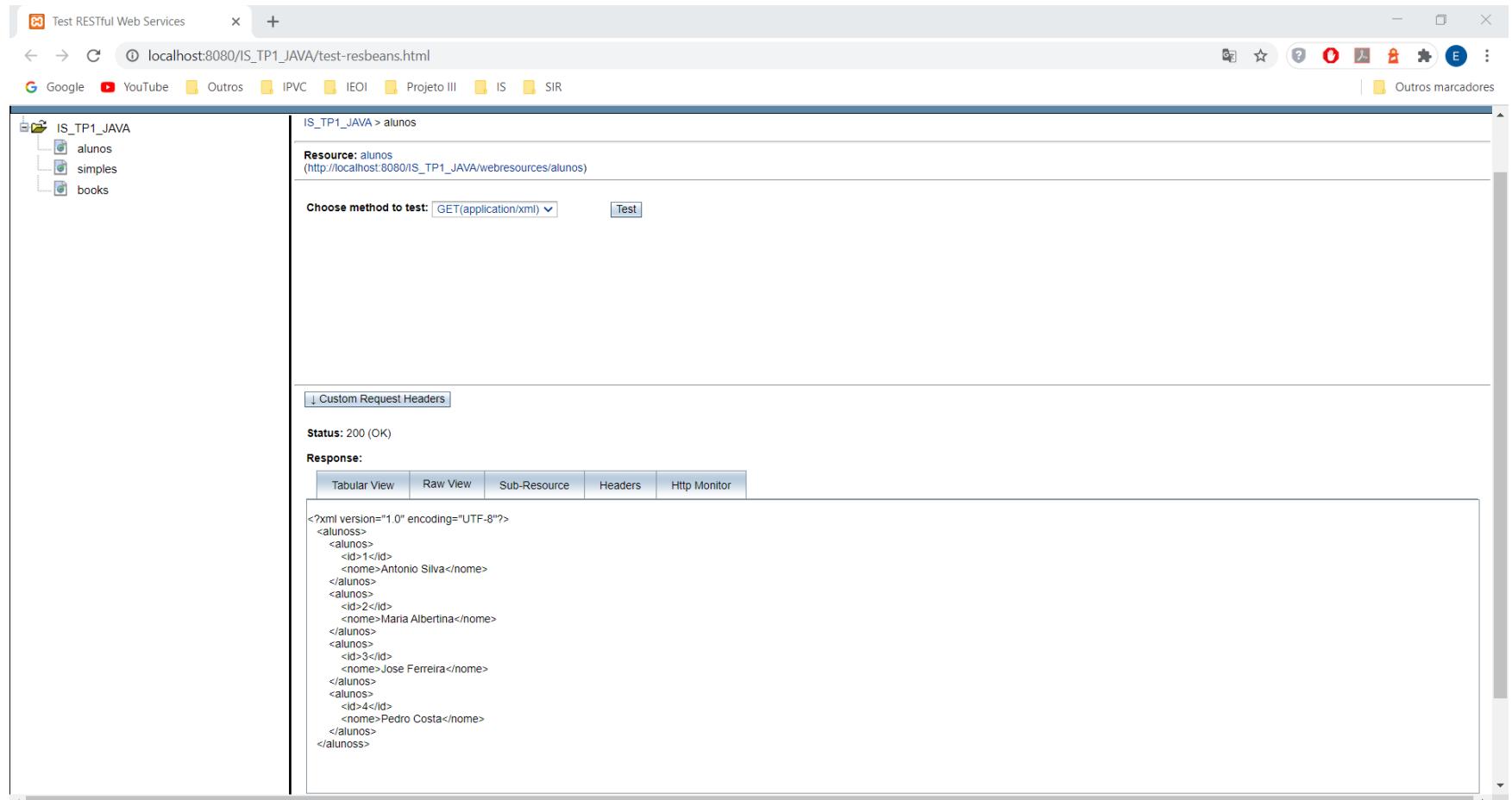
<HTML lang=pt><HEAD><TITLE>Method invocation trace</TITLE></HEAD><H2><A href="#">getAlunos</A></H2>
  
```
- Response Body:**

```

<pre><blockquote><HR><h4>SOAP Response</h4><HR><blockquote><pre xml:lang=&quot;pt&quot;>
<ns2:getAlunos xmlns:ns2="http://db.is.com/">
  <S:Body>
    <ns2:getAlunosResponse xmlns:ns2="http://db.is.com/">
      <id>1</id>
      <nome>Antonio Silva</nome>
    </ns2:getAlunosResponse>
  </S:Body>
</ns2:getAlunos>
</pre></blockquote><HR></pre>
  
```

■ 3.1.6. REST webservice with file alunos.xml from database

Integração de Sistemas



The screenshot shows a browser window titled "Test RESTful Web Services" with the URL "localhost:8080/IS_TP1_JAVA/test-resbeans.html". The left sidebar shows a tree structure with "IS_TP1_JAVA" expanded, containing "alunos", "simples", and "books". The main content area displays the "alunos" resource details. It shows the URL "http://localhost:8080/IS_TP1_JAVA/webresources/alunos". A dropdown menu "Choose method to test:" is set to "GET(application/xml)". Below it is a "Test" button. A "Custom Request Headers" section is present. The "Status" is listed as "200 (OK)". The "Response" section contains tabs for "Tabular View", "Raw View", "Sub-Resource", "Headers", and "Http Monitor". The "Raw View" tab is selected, displaying the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<alunos>
    <id>1</id>
    <nome>Antonio Silva</nome>
</alunos>
<alunos>
    <id>2</id>
    <nome>Maria Albertina</nome>
</alunos>
<alunos>
    <id>3</id>
    <nome>Jose Ferreira</nome>
</alunos>
<alunos>
    <id>4</id>
    <nome>Pedro Costa</nome>
</alunos>
</alunos>
```

■ 3.2. PHP webservices implementation

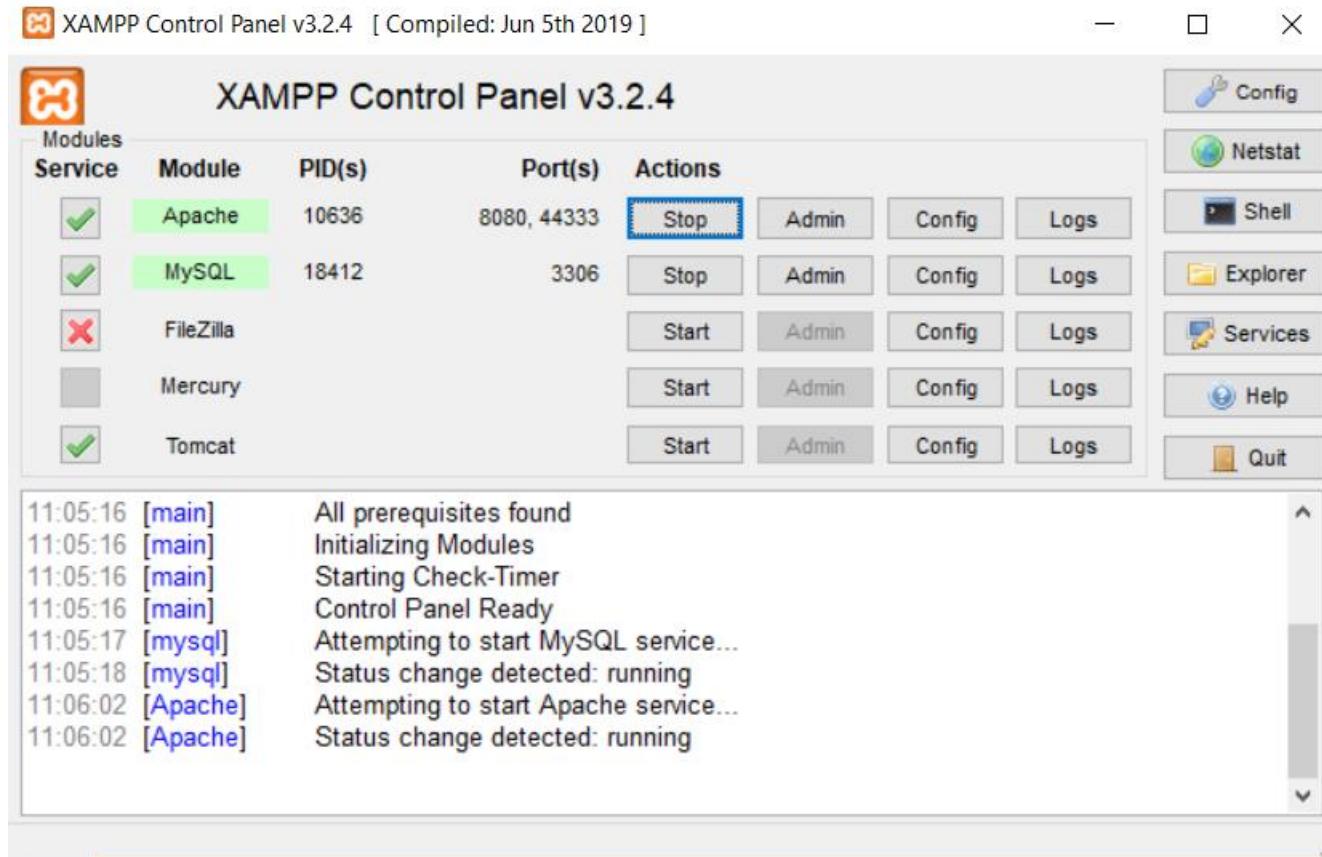
Integração de Sistemas

- For the test of the implemented REST and SOAP webservices, it was started a local Apache server using XAMPP, and the php files accessed via web browser.

- For some of the tests it was necessary to change the values of the variables *username* and *password* in the file *conn.php* which configures the connection to the MySQL database running on the port 3306 in the server localhost, as it is possible to confirm on the next slide.

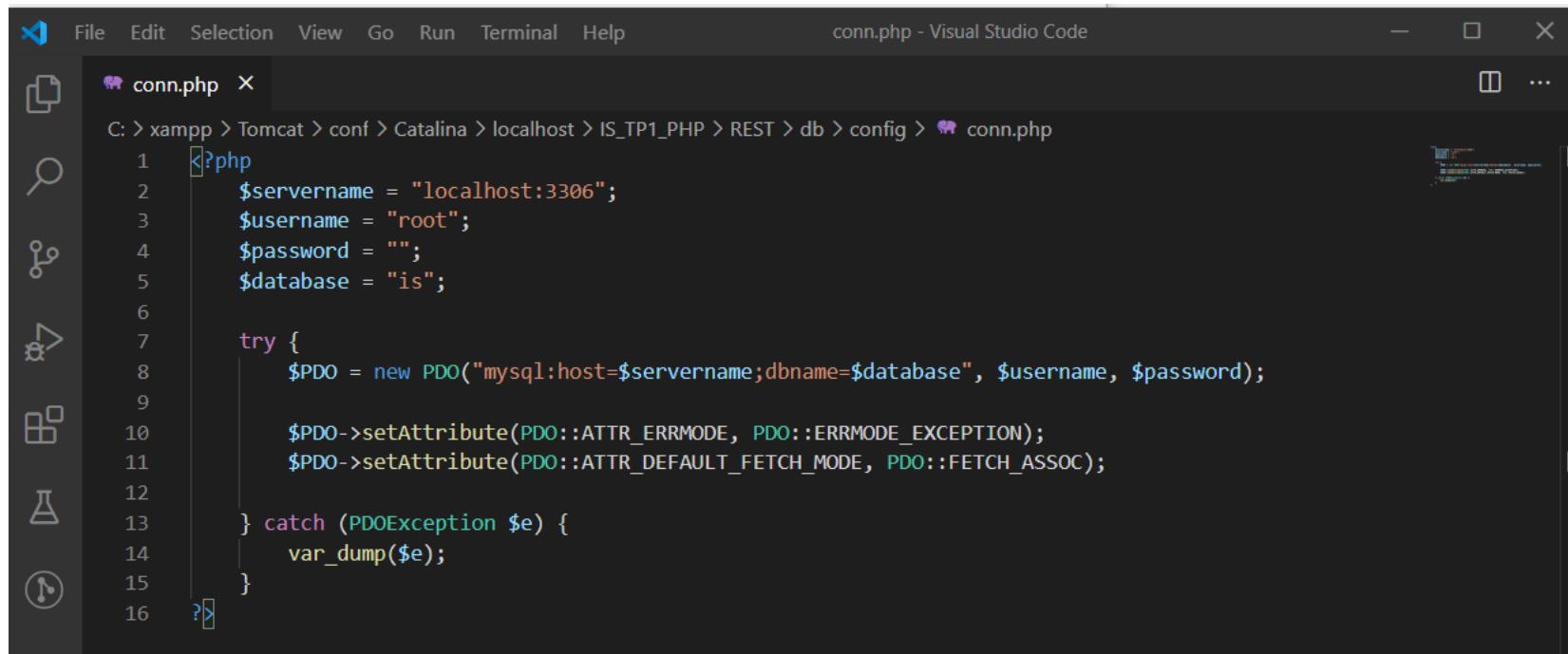
■ 3.2. PHP webservices implementation

Integração de Sistemas



■ 3.2. PHP webservices implementation

Integração de Sistemas



The screenshot shows a Visual Studio Code window with the file "conn.php" open. The code is a PHP script for database connection configuration. It includes variables for servername, username, password, and database, and a try-catch block for PDO exceptions.

```
conn.php - Visual Studio Code
File Edit Selection View Go Run Terminal Help
conn.php ×
C: > xampp > Tomcat > conf > Catalina > localhost > IS_TP1_PHP > REST > db > config > conn.php
1 <?php
2     $servername = "localhost:3306";
3     $username = "root";
4     $password = "";
5     $database = "is";
6
7     try {
8         $PDO = new PDO("mysql:host=$servername;dbname=$database", $username, $password);
9
10        $PDO->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11        $PDO->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
12    } catch (PDOException $e) {
13        var_dump($e);
14    }
15
16 ?>
```

■ 3.2. PHP webservices implementation

Integração de Sistemas

■ The implementation of a REST webservice using PHP, is done by using four different

files:

- One file for the declaration of the class SimpleRest, which implements the REST protocol;
- The second file defines a specific class to get and handle the data from the database or XML file, depending on the example;
- The third, also depending on the example, is optional because it defines and creates the connection to the database using the configuration present in the file *conn.php*;
- The fourth usually contains the class RestController, which uses the specific class (second file) defined to handle the request.

■ 3.2. PHP webservices implementation

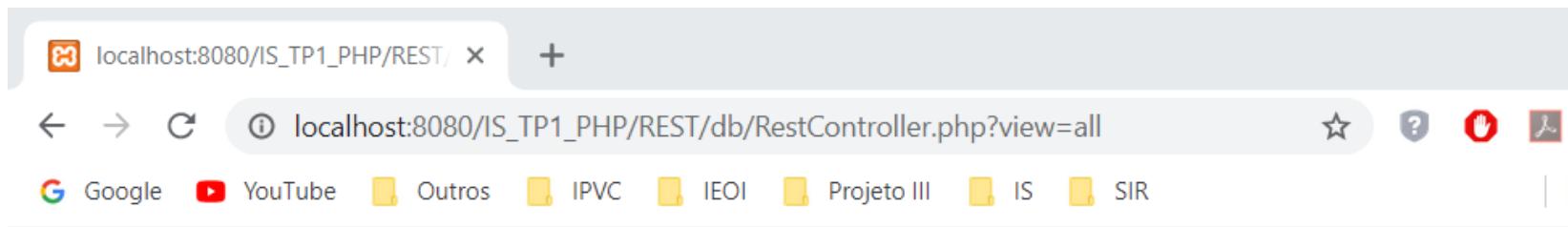
Integração de Sistemas

■ As far as the implementation of webservices using SOAP, it is done with:

- One file for the client, which uses *nusoap* to make a request, in order to get the data and print the results to the user;
- The second file is optional, because it defines and executes the connection to the database, and returns the results;
- The third file defines the server which has the necessary methods to de-serialize the data contained in the XML file or treat the data obtained from the database, depending on the example.

■ 3.2.1. REST webservice to read data form the database

Integração de Sistemas



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<alunos>
  ▼<aluno>
    <id>1</id>
    <nome>Antonio Silva</nome>
  </aluno>
  ▼<aluno>
    <id>2</id>
    <nome>Maria Albertina</nome>
  </aluno>
  ▼<aluno>
    <id>3</id>
    <nome>Jose Ferreira</nome>
  </aluno>
  ▼<aluno>
    <id>4</id>
    <nome>Pedro Costa</nome>
  </aluno>
</alunos>
```

■ 3.2.2. REST webservice using parser DOM/SAX and the books.xml file

Integração de Sistemas

localhost:8080/IS_TP1_PHP/REST

localhost:8080/IS_TP1_PHP/REST/DOM/RestController.php

author: Gambardella, Matthew
title: XML Developer's Guide
genre: Computerprice: 44.95
publish_date: 2000-10-01
description: An in-depth look at creating applications with XML.

author: Ralls, Kim
title: Midnigh Rain
genre: Fantasyprice: 5.95
publish_date: 2000-12-16
description: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

author: Corets, Eva
title: Maeve Ascendant
genre: Fantasyprice: 5.95
publish_date: 2000-11-17
description: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.

author: Corets, Eva
title: Oberon's Legacy
genre: Fantasyprice: 5.95
publish_date: 2001-03-10
description: In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.

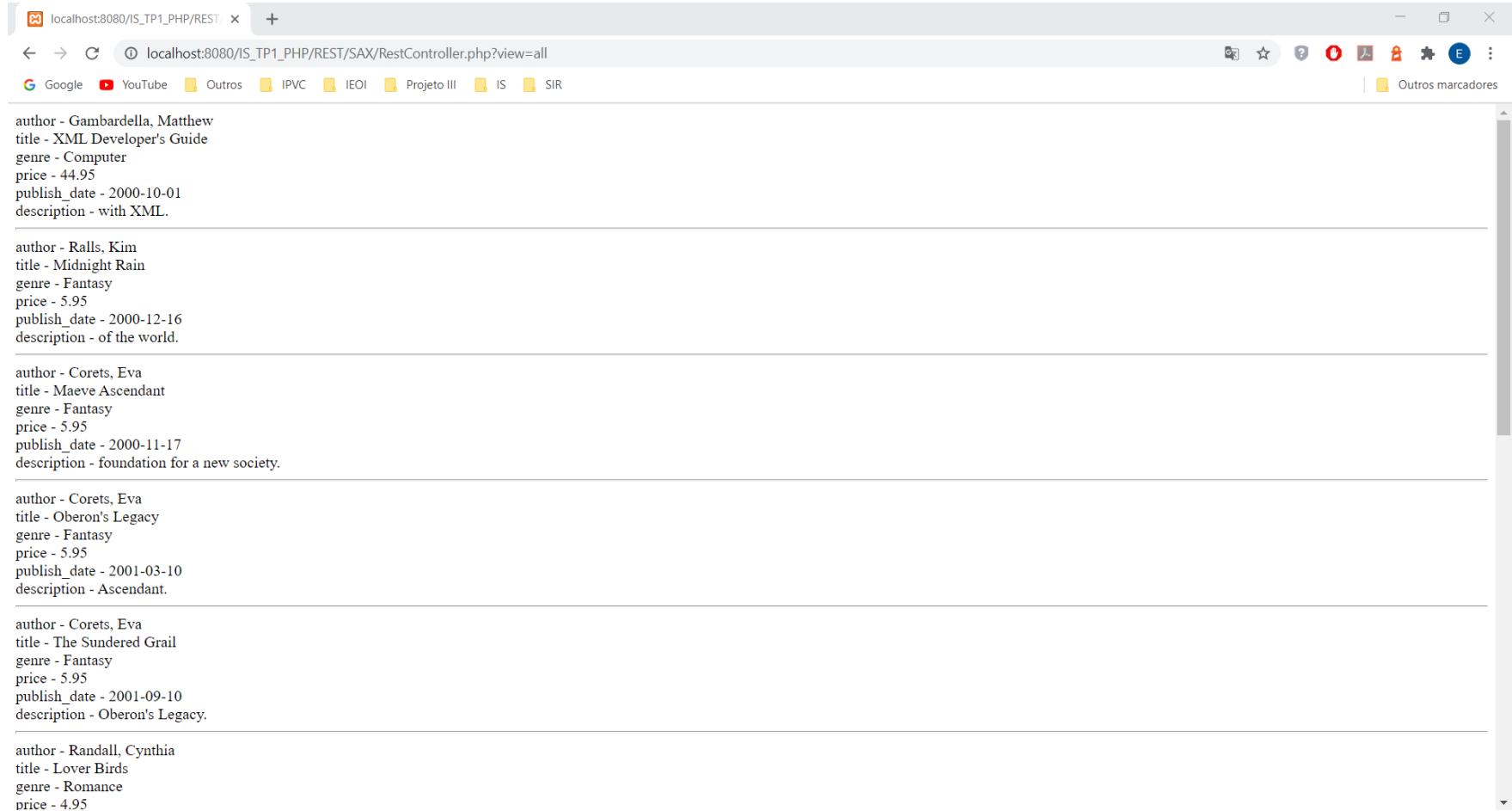
author: Corets, Eva
title: The Sundered Grail
genre: Fantasyprice: 5.95
publish_date: 2001-09-10
description: The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.

author: Randall, Cynthia
title: Lover Birds
genre: Romanceprice: 4.95
publish_date: 2000-09-02
description: When Carla meets Paul at an ornithology conference, tempers fly as feathers get ruffled.

author: Thurman, Paula
title: Splish Splash
genre: Romanceprice: 4.95

■ 3.2.2. REST webservice using parser DOM/SAX and the books.xml file

Integração de Sistemas



localhost:8080/IS_TP1_PHP/REST × +

localhost:8080/IS_TP1_PHP/REST/SAX/RestController.php?view=all

Google YouTube Outros IPVC IEOI Projeto III IS SIR

Outros marcados

author - Gambardella, Matthew
title - XML Developer's Guide
genre - Computer
price - 44.95
publish_date - 2000-10-01
description - with XML.

author - Ralls, Kim
title - Midnight Rain
genre - Fantasy
price - 5.95
publish_date - 2000-12-16
description - of the world.

author - Corets, Eva
title - Maeve Ascendant
genre - Fantasy
price - 5.95
publish_date - 2000-11-17
description - foundation for a new society.

author - Corets, Eva
title - Oberon's Legacy
genre - Fantasy
price - 5.95
publish_date - 2001-03-10
description - Ascendant.

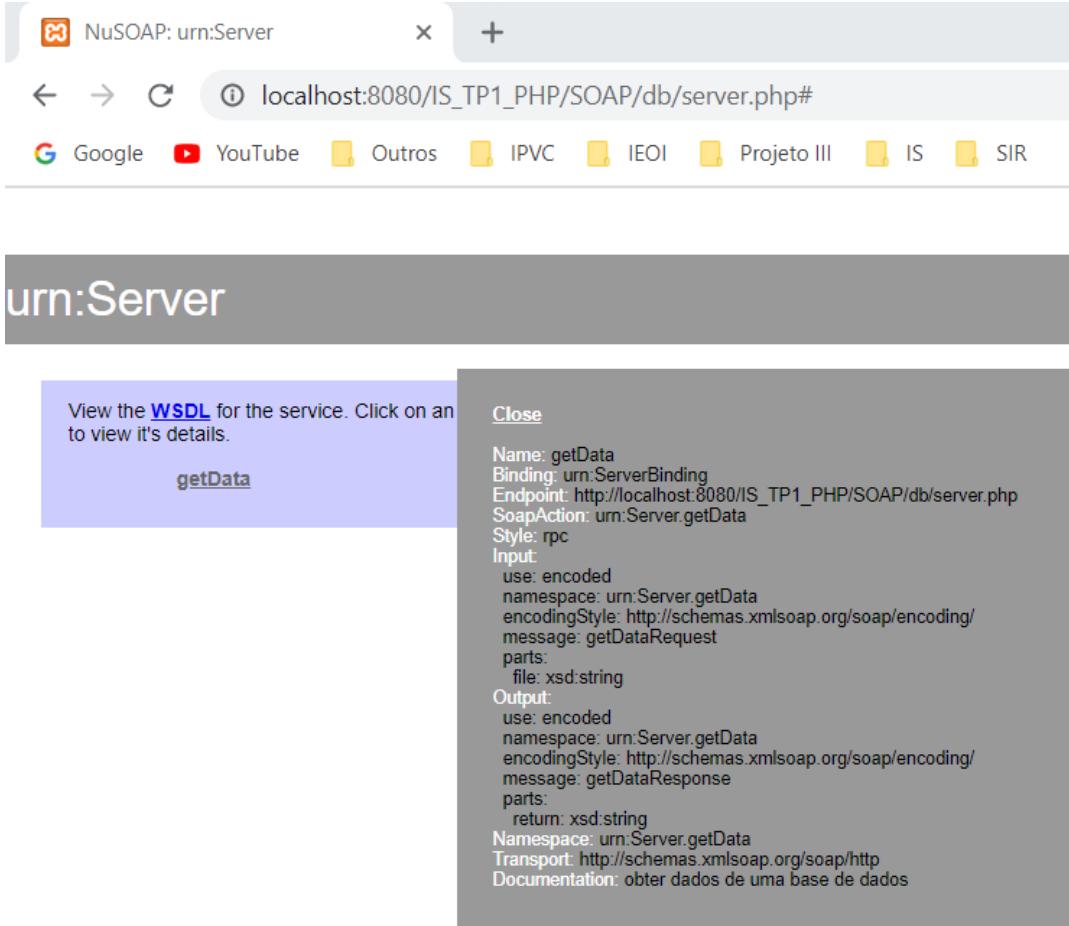
author - Corets, Eva
title - The Sundered Grail
genre - Fantasy
price - 5.95
publish_date - 2001-09-10
description - Oberon's Legacy.

author - Randall, Cynthia
title - Lover Birds
genre - Romance
price - 4.95

■ 3.2.3. SOAP webservice to read data from the database

Integração de Sistemas

■ Server.php execution



NuSOAP: urn:Server

localhost:8080/IS_TP1_PHP/SOAP/db/server.php#

Google YouTube Outros IPVC IEOI Projeto III IS SIR

urn:Server

View the [WSDL](#) for the service. Click on an item to view its details.

[getData](#)

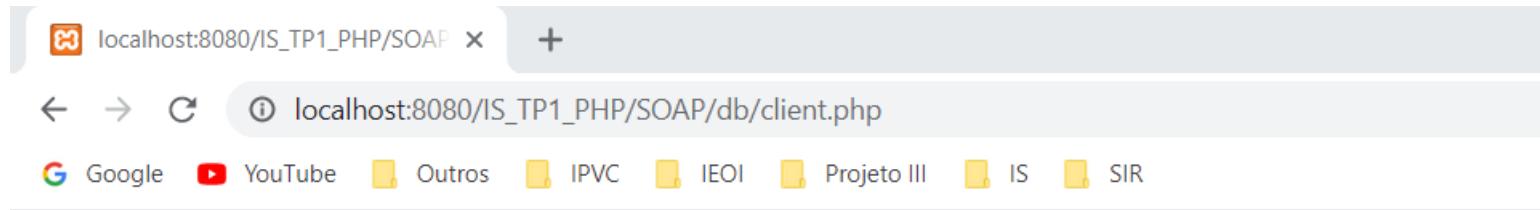
[Close](#)

Name: getData
Binding: urn:ServerBinding
Endpoint: http://localhost:8080/IS_TP1_PHP/SOAP/db/server.php
SoapAction: urn:Server.getData
Style: rpc
Input:
use: encoded
namespace: urn:Server.getData
encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
message: getDataRequest
parts:
file: xsd:string
Output:
use: encoded
namespace: urn:Server.getData
encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
message: getDataResponse
parts:
return: xsd:string
Namespace: urn:Server.getData
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: obter dados de uma base de dados

■ 3.2.3. SOAP webservice to read data from the database

Integração de Sistemas

■ Client.php execution



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<alunos>
  ▼<aluno>
    <id>1</id>
    <nome>Antonio Silva</nome>
  </aluno>
  ▼<aluno>
    <id>2</id>
    <nome>Maria Albertina</nome>
  </aluno>
  ▼<aluno>
    <id>3</id>
    <nome>Jose Ferreira</nome>
  </aluno>
  ▼<aluno>
    <id>4</id>
    <nome>Pedro Costa</nome>
  </aluno>
</alunos>
```

■ 3.2.4. SOAP webservice to read data from the file books.xml using DOM parser

Integração de Sistemas

Screenshot of a web browser showing the output of a SOAP service. The URL is `localhost:8080/IS_TP1_PHP/SOAP/client.php`. The page displays a list of book details from a XML file.

```

author: Gambardella, Matthew
title: XML Developer's Guide
genre: Computerprice: 44.95
publish_date: 2000-10-01
description: An in-depth look at creating applications with XML.

author: Ralls, Kim
title: Midnight Rain
genre: Fantasyprice: 5.95
publish_date: 2000-12-16
description: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

author: Corets, Eva
title: Maeve Ascendant
genre: Fantasyprice: 5.95
publish_date: 2000-11-17
description: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.

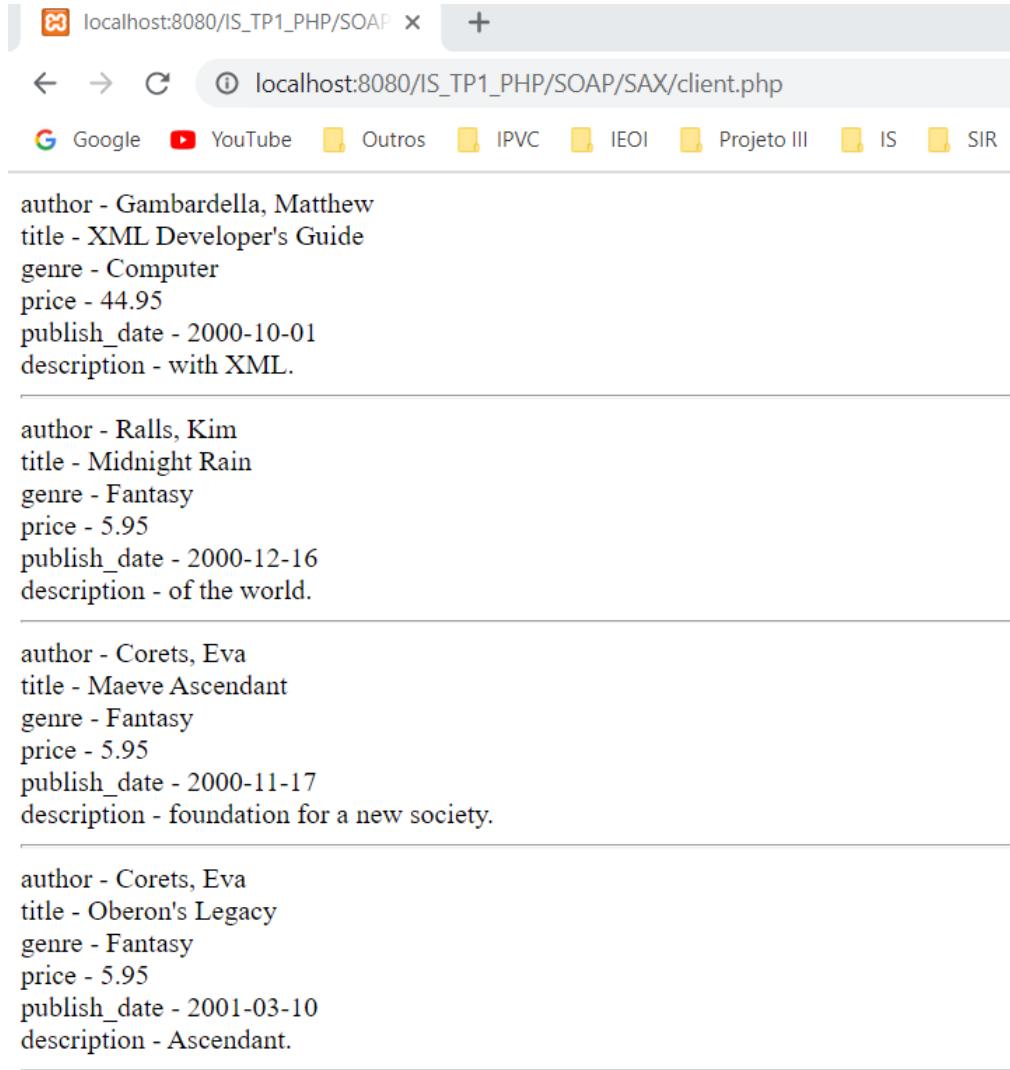
author: Corets, Eva
title: Oberon's Legacy
genre: Fantasyprice: 5.95
publish_date: 2001-03-10
description: In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.

author: Corets, Eva
title: The Sundered Grail
genre: Fantasyprice: 5.95
publish_date: 2001-09-10
description: The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.

author: Randall, Cynthia
title: Lover Birds
genre: Romanceprice: 4.95
publish_date: 2000-09-02
description: When Carla meets Paul at an ornithology conference, tempers fly as feathers get ruffled.

```

■ 3.2.5. SOAP webservice to read data from the file books.xml using SAX parser



The screenshot shows a web browser window with the URL `localhost:8080/IS_TP1_PHP/SOAP/client.php`. The page displays four sets of book data extracted from an XML file using a SAX parser. Each set includes author, title, genre, price, publish_date, and description.

author - Gambardella, Matthew
title - XML Developer's Guide
genre - Computer
price - 44.95
publish_date - 2000-10-01
description - with XML.

author - Ralls, Kim
title - Midnight Rain
genre - Fantasy
price - 5.95
publish_date - 2000-12-16
description - of the world.

author - Corets, Eva
title - Maeve Ascendant
genre - Fantasy
price - 5.95
publish_date - 2000-11-17
description - foundation for a new society.

author - Corets, Eva
title - Oberon's Legacy
genre - Fantasy
price - 5.95
publish_date - 2001-03-10
description - Ascendant.

■ 4. Output correction in client side

- In this section, the code in the cliente side for both the SOAP and REST web services implemented in PHP, was altered in order to present the result of the parse of the books.xml file, not in html format but in XML. The code and result can be examined in the next two points.

■ 4.1 REST webservice with the file books.xml and parser DOM

Integração de Sistemas

■ In this file the only changes made were:

- Commented all the code beneath line 7;
- Added the code on line two, indicating that the php file will return XML;
- Instead of getting the content by its elements and building a string to send to the user, the file is saved as XML and this action printed back to the user.

```

RestController.php BooksRestHandler.php readFile.php X

C: > xampp > htdocs > IS_TP1_PHP > REST > DOM > readFile.php

1  <?php
2  header('content-type: application/xml');
3  $dom = new domDocument;
4
5  $dom->load("files/books.xml");
6
7  echo $dom->saveXML[0];
//$dom->preserveWhiteSpace = false;
8
9
10 //tables = $dom->getElementsByTagName('catalog');
11
12 //rows = $tables->item(0)->getElementsByTagName('book');
13
14 /*
15 foreach ($rows as $row) {
16
17     $author = $row->getElementsByTagName('author');
18     $title = $row->getElementsByTagName('title');
19     $genre = $row->getElementsByTagName('genre');
20     $price = $row->getElementsByTagName('price');
21     $publish_date = $row->getElementsByTagName('publish_date');
22     $description = $row->getElementsByTagName('description');
23
24
25     echo 'author: '.$author->item(0)->nodeValue.'<br />';
26     echo 'title: '.$title->item(0)->nodeValue.'<br />';
27     echo 'genre: '.$genre->item(0)->nodeValue;
28     echo 'price: '.$price->item(0)->nodeValue.'<br />';
29     echo 'publish_date: '.$publish_date->item(0)->nodeValue.'<br />';
30     echo 'description: '.$description->item(0)->nodeValue;
31     echo '<hr />';
32 }
33 */
34 ?>

```

■ 4.1 REST webservice with the file books.xml and parser DOM

Integração de Sistemas

■ Obtained result:

Screenshot of a web browser showing the XML document tree for 'books.xml'.

The browser address bar shows: localhost:8080/IS_TP1_PHP/REST/RestController.php

The page content displays the XML document structure:

```

<catalog>
  <book id="bk101">
    <author>Gambardea, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.</description>
  </book>
  <book id="bk103">
    <author>Corets, Eva</author>
    <title>Maeve Ascendant</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-11-17</publish_date>
    <description>After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.</description>
  </book>
  <book id="bk104">
    <author>Corets, Eva</author>
    <title>Oberon's Legacy</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-03-10</publish_date>
    <description>In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.</description>
  </book>
  <book id="bk105">
    <author>Corets, Eva</author>
    <title>The Sundered Grail</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-09-10</publish_date>
    <description>The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.</description>
  </book>
  <book id="bk106">
    <author>Randall, Cynthia</author>
    <title>Lover Birds</title>
  </book>

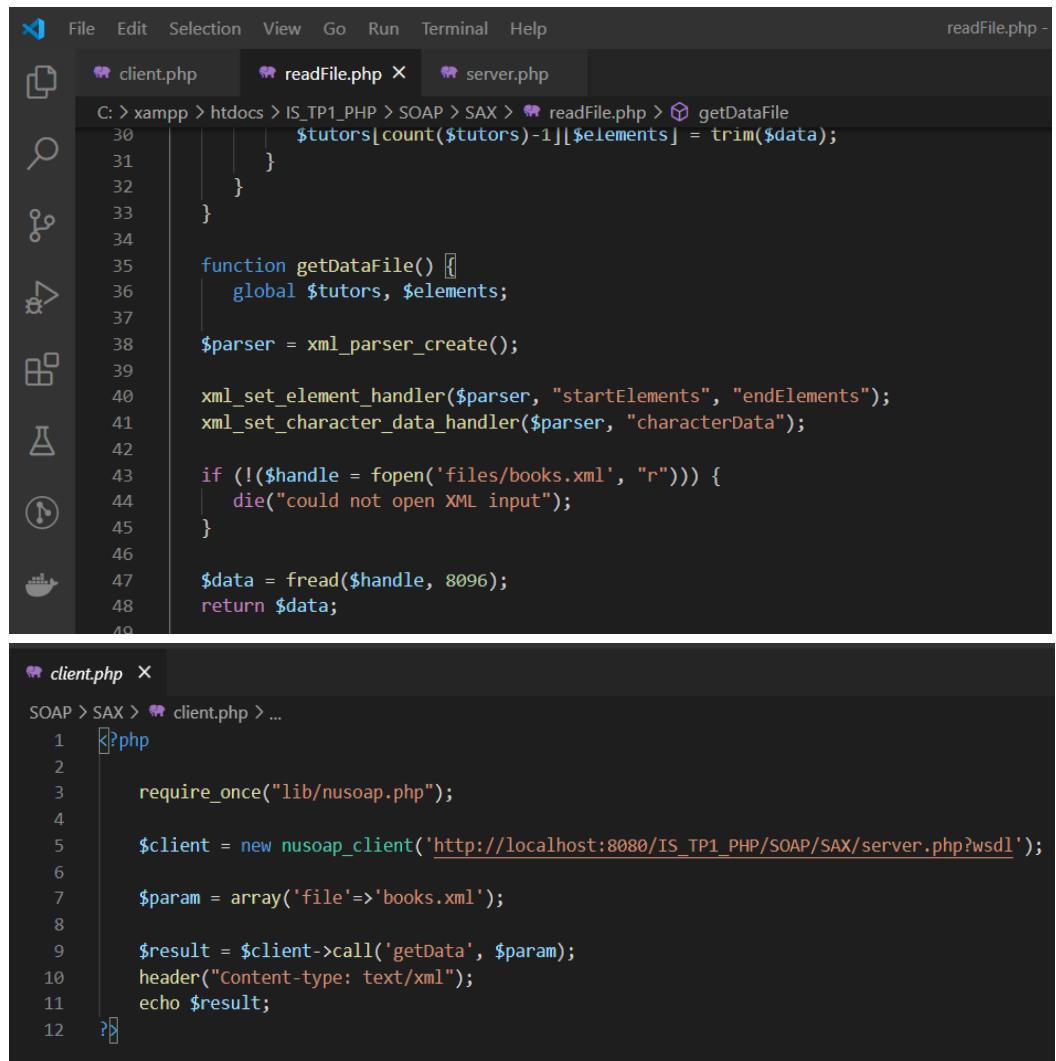
```

■ 4.2 SOAP webservice with the file books.xml and parser SAX

Integração de Sistemas

■ In this file the changes include:

- Commented all the code beneath line 48;
- The file opened in line 43, has his data fully extracted and directly printed to the client;
- Added line 10 on the client.php file to indicate that the data received is XML;



The screenshot shows a code editor with two tabs open: "readFile.php" and "client.php".

readFile.php

```

File Edit Selection View Go Run Terminal Help
client.php  readFile.php X server.php
C: > xampp > htdocs > IS_TP1_PHP > SOAP > SAX > readFile.php > getaDataFile
30     $tutors[$count($tutors)-1][$elements] = trim($data);
31 }
32 }
33 }
34
35 function getaDataFile() {
36     global $tutors, $elements;
37
38     $parser = xml_parser_create();
39
40     xml_set_element_handler($parser, "startElements", "endElements");
41     xml_set_character_data_handler($parser, "characterData");
42
43     if (!$handle = fopen('files/books.xml', "r")) {
44         die("could not open XML input");
45     }
46
47     $data = fread($handle, 8096);
48     return $data;
49

```

client.php

```

SOAP > SAX > client.php ...
1 <?php
2
3     require_once("lib/nusoap.php");
4
5     $client = new nusoap_client('http://localhost:8080/IS_TP1_PHP/SOAP/SAX/server.php?wsdl');
6
7     $param = array('file'=>'books.xml');
8
9     $result = $client->call('getaData', $param);
10    header("Content-type: text/xml");
11    echo $result;
12 ?>

```

■ 4.2 SOAP webservice with the file books.xml and parser SAX

Integração de Sistemas

■ Obtained result:

Screenshot of a web browser showing the XML document tree for 'books.xml'.

The address bar shows: localhost:8080/IS_TP1_PHP/SOAP/SAX/client.php

The page content starts with: "This XML file does not appear to have any style information associated with it. The document tree is shown below."

```
<catalog>
  <book id="bk101">
    <author>Gambardea, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.</description>
  </book>
  <book id="bk103">
    <author>Corets, Eva</author>
    <title>Maeve Ascendant</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-11-17</publish_date>
    <description>After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.</description>
  </book>
  <book id="bk104">
    <author>Corets, Eva</author>
    <title>Oberon's Legacy</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-03-10</publish_date>
    <description>In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.</description>
  </book>
  <book id="bk105">
    <author>Corets, Eva</author>
    <title>The Sundered Grail</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-09-10</publish_date>
    <description>The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.</description>
  </book>
  <book id="bk106">
    <author>Randall, Cynthia</author>
    <title>Lover Birds</title>
```

■ 5 Publish a Java project in a Tomcat server

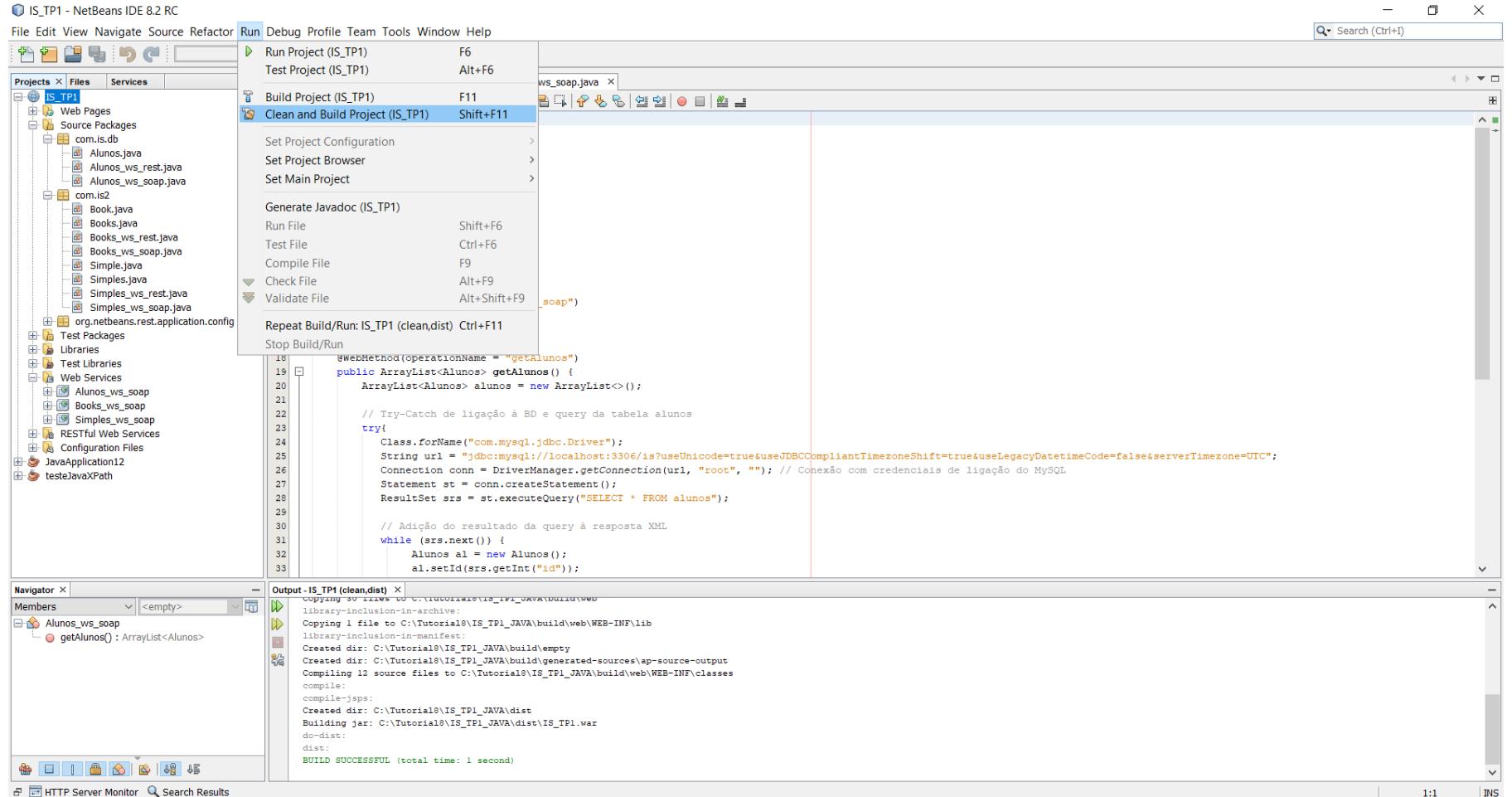
Integração de Sistemas

- In this part of the practical work, the main objective is the generation of a *war* file from an existing NetBeans web project. This project includes the code for some SOAP and REST web services implemented in Java, and the purpose of publishing it is to place these web services in the server without allowing access to the source code, functioning outside of the development environment.

- The first step of this process is to clean and build the project, this will create a war file that can be uploaded in the Tomcat server web application manager, the second step is to select the war file and hit the deploy button and the final step is to check the web application manager and see if the project is deployed in the server.

■ 5 Publish a Java project in a Tomcat server

Integração de Sistemas

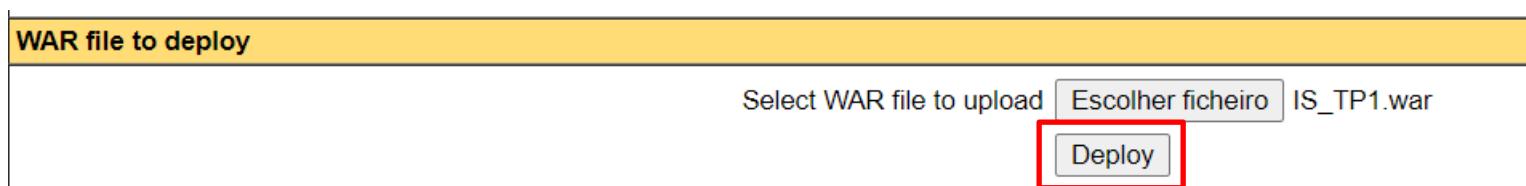
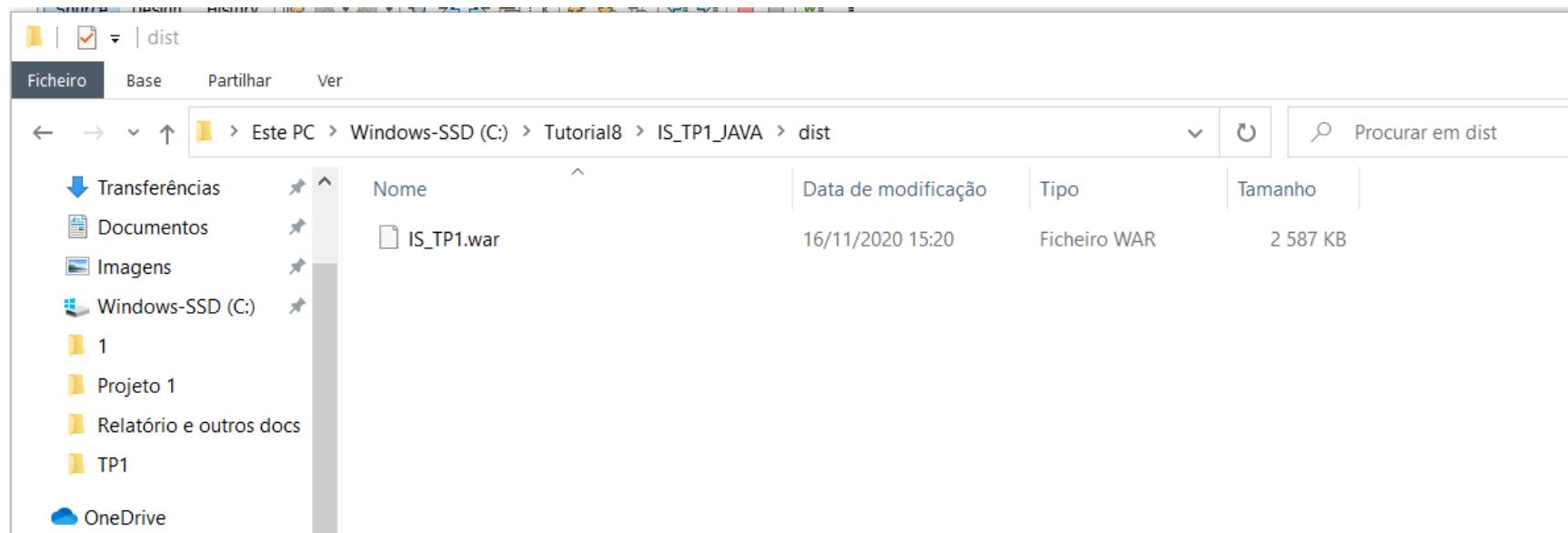


The screenshot shows the NetBeans IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer:** Projects (IS_TP1), Files, Services. The IS_TP1 project contains several source packages like com.is.db, com.is2, and org.netbeans.rest.application.config, along with test packages, libraries, and web services.
- Code Editor:** The ws_soap.java file contains Java code for a SOAP service. It includes annotations like @WebMethod(operationName = "getAlunos") and @RequestWrapper(localName = "getAlunos"). The code implements the getAlunos() method which queries a MySQL database for student records.
- Output Window:** Shows the build logs for the IS_TP1 project. It includes commands like "Copying 30 files to C:\Tutorial8\IS_TP1_JAVA\build\web\WEB-INF\lib\library-inclusion-in-archive:", "Copying 1 file to C:\Tutorial8\IS_TP1_JAVA\build\web\WEB-INF\lib\library-inclusion-in-manifest:", and "BUILD SUCCESSFUL (total time: 1 second)".
- Bottom Bar:** Includes icons for file operations like Open, Save, Find, and Build, along with tabs for HTTP Server Monitor and Search Results.

■ 5 Publish a Java project in a Tomcat server

Integração de Sistemas



WAR file to deploy

Select WAR file to upload IS_TP1.war

■ 5 Publish a Java project in a Tomcat server

Integração de Sistemas

localhost:8080/manager/html/upload?org.apache.catalina.filters.CSRF_NONCE=07EE3315E31481D7C78D1FA203BEAA2D

Google YouTube Outros IPVC IEOI Projeto III IS SIR TV SAMSUNG QE6... Soundbar SONY HT... MEGA Deployment of an E... Outros marcadores

Tomcat Web Application Manager

Message: OK

Manager					
List Applications	HTML Manager Help			Manager Help	Server Status
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/IS_TP1	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

■ 6 XPATH exploration in XML files

Integração de Sistemas

- The XPATH exploration was done by changing the code on two files, the first one was the REST web service with parser JAXB and the file books.xml in Java, and the second one the SOAP web service using the parser DOM and the file books.xml in PHP.

- The changes made were the implementation of XPATH, using expressions to filter through the existing books in the referenced file, only obtaining the books where the author value was equal to “Corets, Eva” and “Thurman, Paula”.

■ 6.1 REST webservice with the file books.xml and parser JAXB - Java

■ Java code with necessary changes made

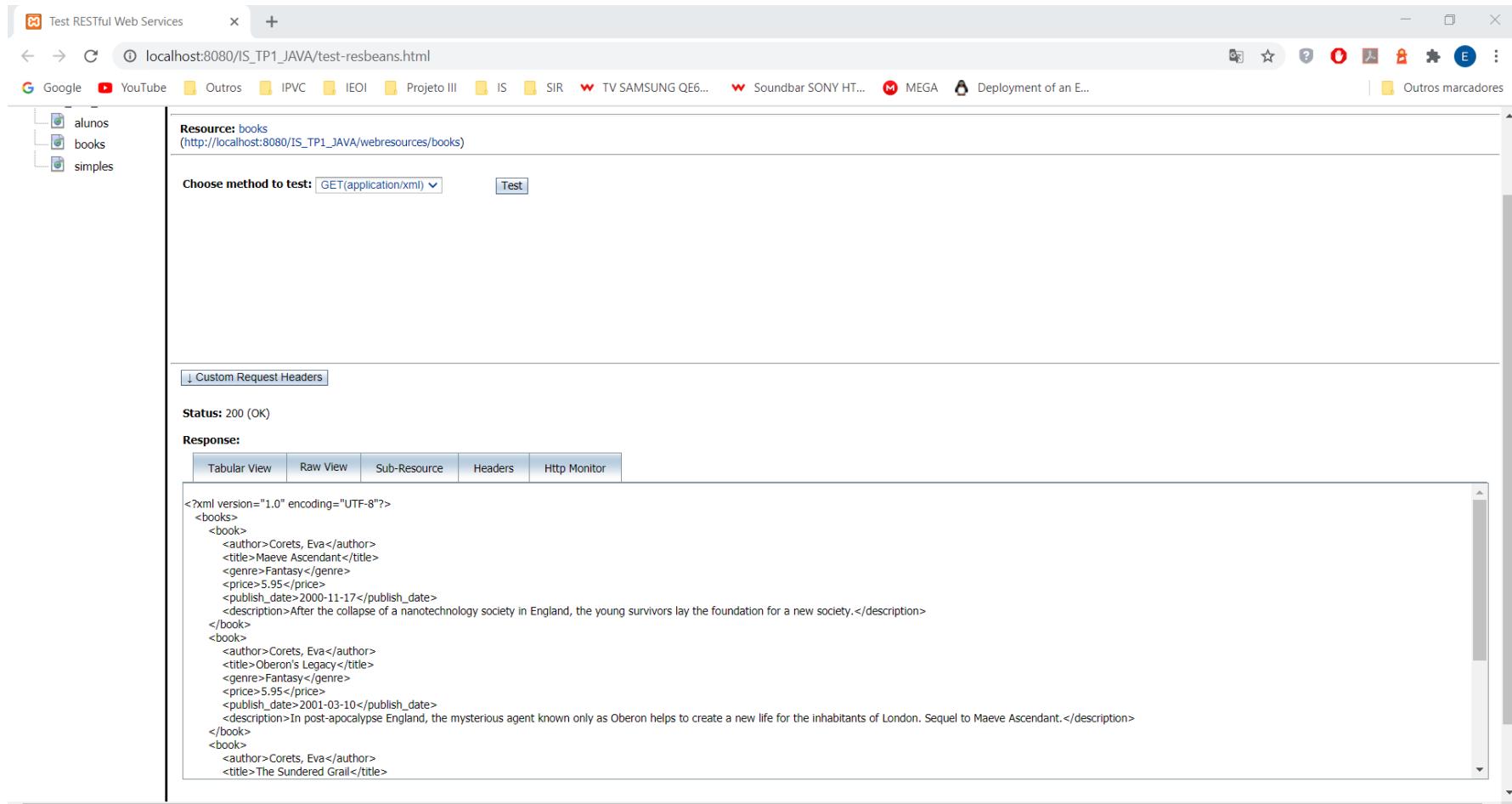
The screenshot shows an IDE interface with multiple tabs at the top: 'Books_ws_rest.java', 'Book.java', and 'Alunos_ws_soap.java'. The main window displays Java code for reading an XML file named 'books.xml' located at 'C:\Tutorial8\FicheirosXML\books.xml'. The code uses JAXBContext and DocumentBuilderFactory to parse the XML and extract book titles. It then filters the books based on specific authors ('Corets, Eva' or 'Thurman, Paula') and returns a list of books with titles matching the filter.

```
34     @Produces(MediaType.APPLICATION_XML) // Anotação que permite que a ArrayList seja devolvida em formato XML
35     public ArrayList<Book> getBooks() throws JAXBException, InvocationTargetException, XPathExpressionException, ParserConfigurationException, SAXException, IOException {
36
37     // Try-Catch do parsing via JAXB do ficheiro
38     try {
39         JAXBContext jaxbContext; // Parser usado (JAXB)
40         File xmlFile = new File("C:\\Tutorial8\\FicheirosXML\\books.xml"); // Cria variável de ficheiro local indicando o path do ficheiro "books.xml"
41         jaxbContext = JAXBContext.newInstance(Books.class); // Cria instance do parse do JAXB na classe da lista de livros
42         Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller(); // Cria interface responsável para "des-serializar" o ficheiro XML para objeto Java
43         Books books = (Books) jaxbUnmarshaller.unmarshal(xmlFile); // Des-serializa o ficheiro XML para objeto Java
44
45
46         DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
47         DocumentBuilder dBuilder;
48
49         dBuilder = dbFactory.newDocumentBuilder();
50
51         Document doc = dBuilder.parse(xmlFile);
52         doc.getDocumentElement().normalize();
53
54         XPath xPath = XPathFactory.newInstance().newXPath();
55
56         String expression = "/catalog/book[author='Corets, Eva' or author='Thurman, Paula']";
57         NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(
58             doc, XPathConstants.NODESET);
59
60         ArrayList<String> booksToKeep = new ArrayList<>();
61         for (int i = 0; i < nodeList.getLength(); i++) {
62             Node nNode = nodeList.item(i);
63
64             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
65                 Element eElement = (Element) nNode;
66                 booksToKeep.add(eElement.getElementsByTagName("title").item(0).getTextContent());
67             }
68         }
69
70         ArrayList<Book> listaBooksFiltrada = new ArrayList<>();
71         for(String title : booksToKeep) {
72             for(Book book : books.getBooks()) {
73                 if(book.getTitle().equals(title)) {
74                     listaBooksFiltrada.add(book);
75                 }
76             }
77         }
78
79
80         return listaBooksFiltrada;
81     }
```

■ 6.1 REST webservice with the file books.xml and parser JAXB - Java

Integração de Sistemas

■ Web service test and results



Resource: books
(http://localhost:8080/IS_TP1_JAVA/webresources/books)

Choose method to test:

Status: 200 (OK)

Response:

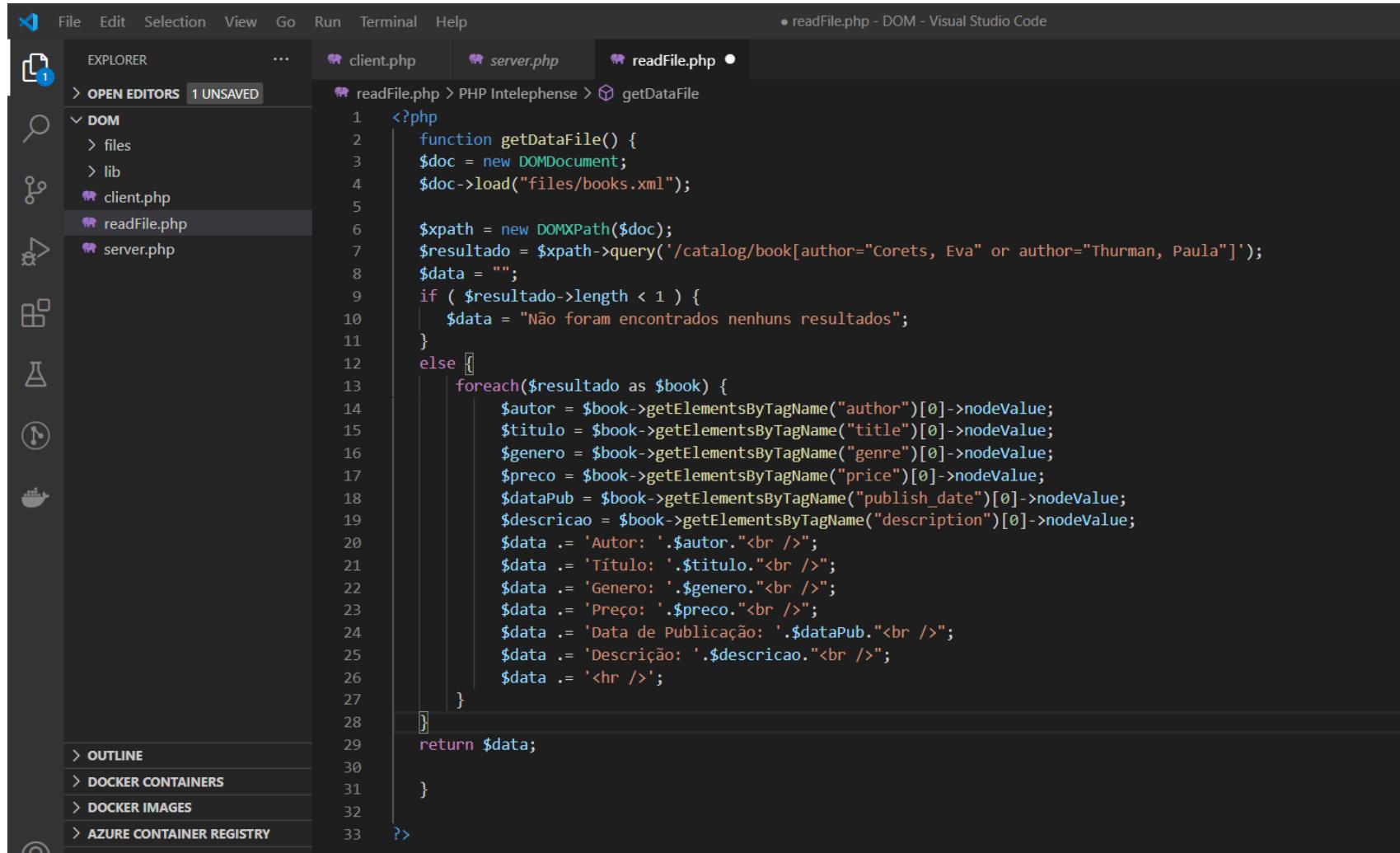
<input type="button" value="Tabular View"/>	<input type="button" value="Raw View"/>	<input type="button" value="Sub-Resource"/>	<input type="button" value="Headers"/>	<input type="button" value="Http Monitor"/>
---	---	---	--	---

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
  <book>
    <author>Corets, Eva</author>
    <title>Maeve Ascendant</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-11-17</publish_date>
    <description>After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.</description>
  </book>
  <book>
    <author>Corets, Eva</author>
    <title>Oberon's Legacy</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-03-10</publish_date>
    <description>In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.</description>
  </book>
  <book>
    <author>Corets, Eva</author>
    <title>The Sundered Grail</title>
```

■ 6.2 SOAP webservice with the file books.xml and parser DOM - PHP

Integração de Sistemas

■ PHP with changes applied



```

File Edit Selection View Go Run Terminal Help
• readfile.php - DOM - Visual Studio Code

EXPLORER client.php server.php readfile.php
OPEN EDITORS 1 UNSAVED
DOM
  files
  lib
  client.php
  readfile.php
  server.php

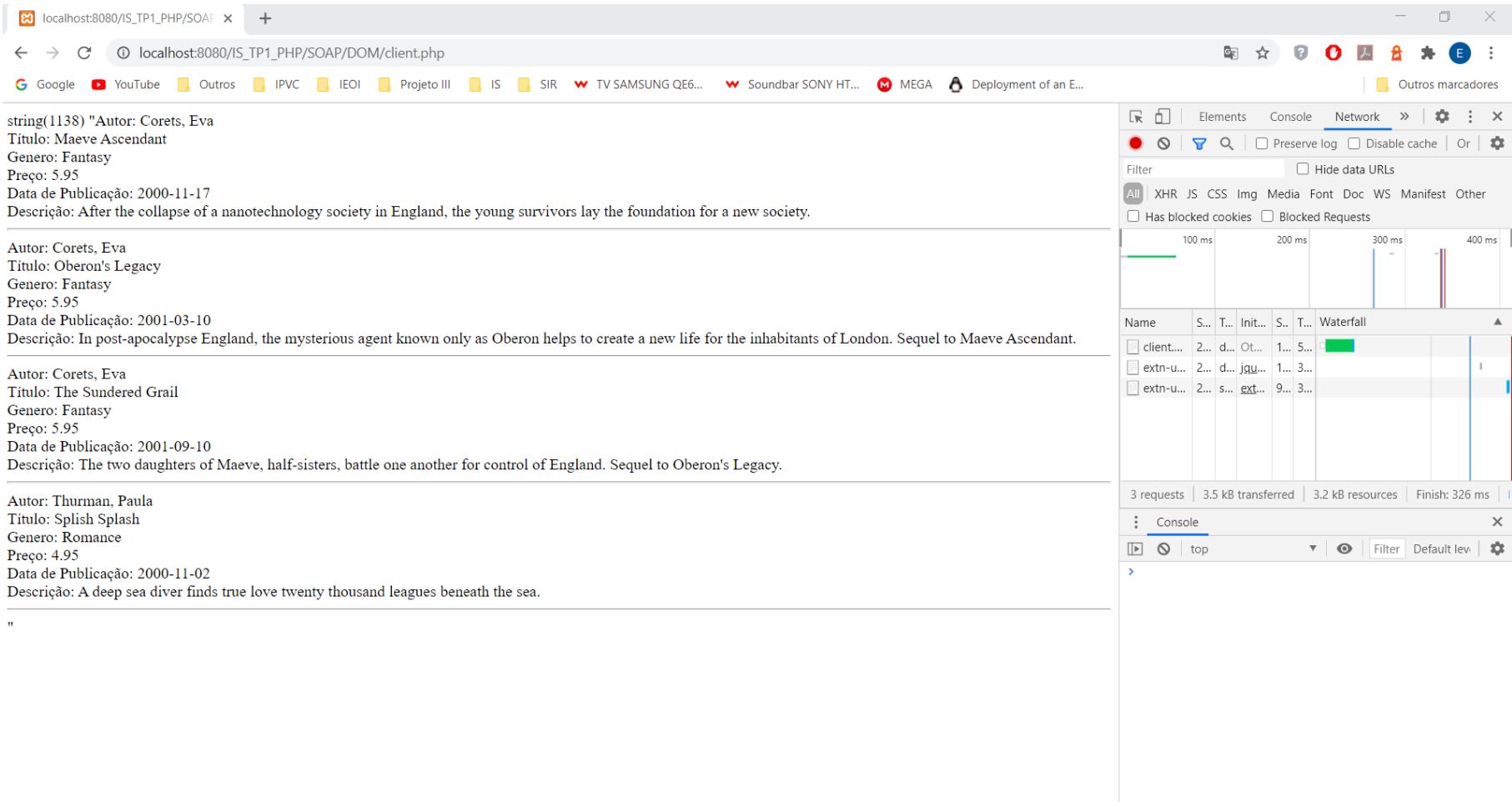
readfile.php > PHP Intelephense > getDataFile
1 <?php
2     function getDataFile() {
3         $doc = new DOMDocument;
4         $doc->load("files/books.xml");
5
6         $xpath = new DOMXPath($doc);
7         $resultado = $xpath->query('/catalog/book[author="Corets, Eva" or author="Thurman, Paula"]');
8         $data = "";
9         if ( $resultado->length < 1 ) {
10             $data = "Não foram encontrados nenhuns resultados";
11         }
12         else {
13             foreach($resultado as $book) {
14                 $autor = $book->getElementsByTagName("author")[0]->nodeValue;
15                 $titulo = $book->getElementsByTagName("title")[0]->nodeValue;
16                 $genero = $book->getElementsByTagName("genre")[0]->nodeValue;
17                 $preco = $book->getElementsByTagName("price")[0]->nodeValue;
18                 $dataPub = $book->getElementsByTagName("publish_date")[0]->nodeValue;
19                 $descricao = $book->getElementsByTagName("description")[0]->nodeValue;
20                 $data .= 'Autor: '.$autor."<br />";
21                 $data .= 'Título: '.$titulo."<br />";
22                 $data .= 'Genero: '.$genero."<br />";
23                 $data .= 'Preço: '.$preco."<br />";
24                 $data .= 'Data de Publicação: '.$dataPub."<br />";
25                 $data .= 'Descrição: '.$descricao."<br />";
26                 $data .= '<hr />';
27             }
28         }
29     return $data;
30 }
31 ?>

```

■ 6.2 SOAP webservice with the file books.xml and parser DOM - PHP

Integração de Sistemas

■ PHP obtained result



localhost:8080/IS_TP1_PHP/SOAP × +

localhost:8080/IS_TP1_PHP/SOAP/DOM/client.php

Google YouTube Outros IPVC IEOI Projeto III IS SIR TV SAMSUNG QE6... Soundbar SONY HT... MEGA Deployment of an E...

Outros marcadores

string(1138) "Autor: Corets, Eva
Título: Maeve Ascendant
Genero: Fantasy
Preço: 5.95
Data de Publicação: 2000-11-17
Descrição: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.

Autor: Corets, Eva
Título: Oberon's Legacy
Genero: Fantasy
Preço: 5.95
Data de Publicação: 2001-03-10
Descrição: In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.

Autor: Corets, Eva
Título: The Sundered Grail
Genero: Fantasy
Preço: 5.95
Data de Publicação: 2001-09-10
Descrição: The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.

Autor: Thurman, Paula
Título: Splish Splash
Genero: Romance
Preço: 4.95
Data de Publicação: 2000-11-02
Descrição: A deep sea diver finds true love twenty thousand leagues beneath the sea.

"

Elements Console Network Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Has blocked cookies Blocked Requests

100 ms 200 ms 300 ms 400 ms

Name	S...	T...	Init...	S...	T...	Waterfall
client...	2...	d...	Ot...	1...	5...	█
extrn-u...	2...	d...	jgu...	1...	3...	█
extrn-u...	2...	s...	ext...	9...	3...	█

3 requests | 3.5 kB transferred | 3.2 kB resources | Finish: 326 ms

Console

top Filter Default lev

■ 7 Submitting large XML files

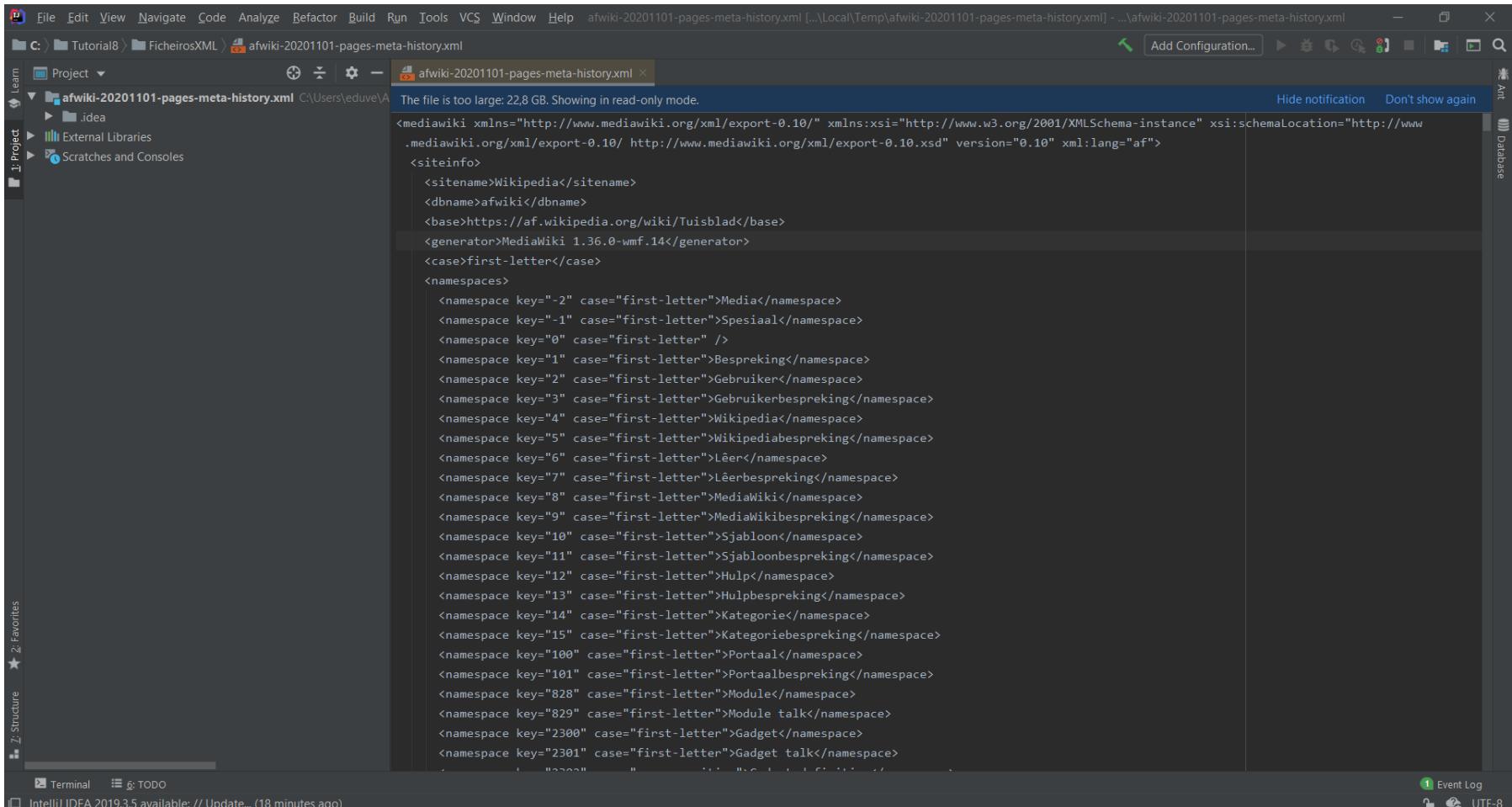
Integração de Sistemas

- This section demonstrates the attempt of submitting and parsing a large XML file, in this case with 485 MB, to different web services, in Java and PHP. This file was downloaded from <https://dumps.wikimedia.org/mirrors.html>, and contains several types of data from multiple Wikipedia pages.
- This will be accomplished by changing the code in the REST web service and parser JAXB, in Java to import the new file and create the necessary classes that will be the model of the file, allowing to convert the file into a Java object. In the SOAP web service and parser DOM in PHP, it was only changed the code to import the new file, parsing and printing its contents to the client.

■ 7 Submitting large XML files

Integração de Sistemas

■ afwiki-20201101-pages-articles.xml file structure



```

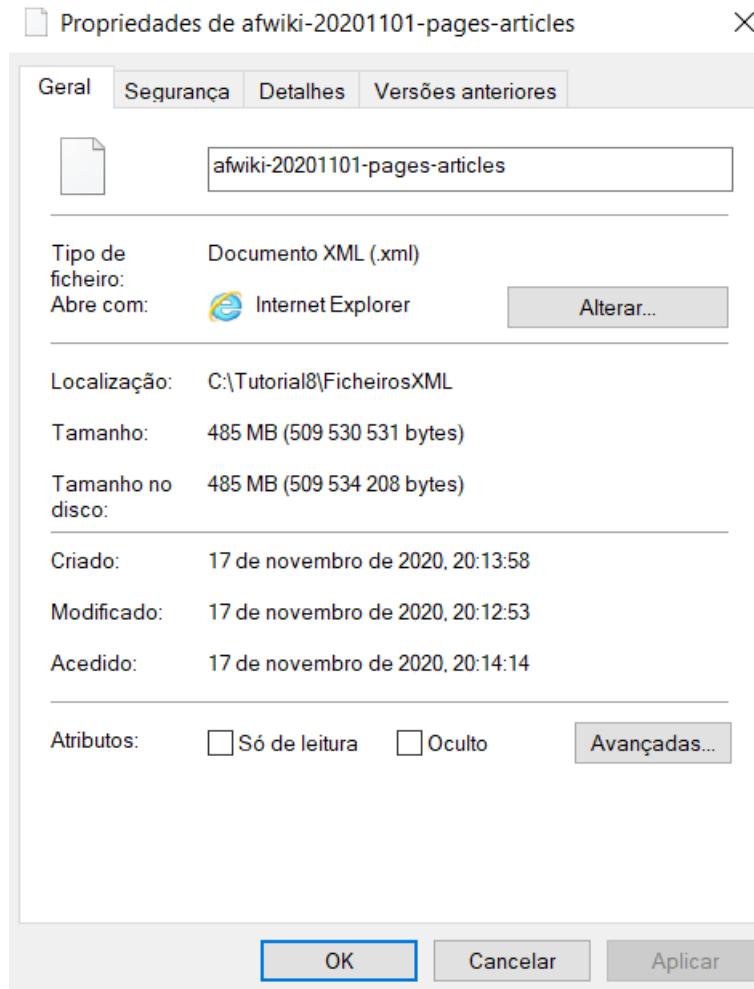
<mediawiki xmlns="http://www.mediawiki.org/xml/export-0.10/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.mediawiki.org/xml/export-0.10/ http://www.mediawiki.org/xml/export-0.10.xsd" version="0.10" xml:lang="af">
<siteinfo>
  <sitename>Wikipedia</sitename>
  <dbname>afwiki</dbname>
  <base>https://af.wikipedia.org/wiki/Tuisblad</base>
  <generator>MediaWiki 1.36.0-wmf.14</generator>
  <case>first-letter</case>
<namespaces>
  <namespace key="-2" case="first-letter">Media</namespace>
  <namespace key="-1" case="first-letter">Spesiaal</namespace>
  <namespace key="0" case="first-letter" />
  <namespace key="1" case="first-letter">Besprekking</namespace>
  <namespace key="2" case="first-letter">Gebruiker</namespace>
  <namespace key="3" case="first-letter">Gebruikerbesprekking</namespace>
  <namespace key="4" case="first-letter">Wikipedia</namespace>
  <namespace key="5" case="first-letter">Wikipediabesprekking</namespace>
  <namespace key="6" case="first-letter">Lêer</namespace>
  <namespace key="7" case="first-letter">Lêerbesprekking</namespace>
  <namespace key="8" case="first-letter">MediaWiki</namespace>
  <namespace key="9" case="first-letter">MediaWikibesprekking</namespace>
  <namespace key="10" case="first-letter">Sjabloon</namespace>
  <namespace key="11" case="first-letter">Sjabloonbesprekking</namespace>
  <namespace key="12" case="first-letter">Hulp</namespace>
  <namespace key="13" case="first-letter">Hulpbesprekking</namespace>
  <namespace key="14" case="first-letter">Kategorie</namespace>
  <namespace key="15" case="first-letter">Kategoriebesprekking</namespace>
  <namespace key="100" case="first-letter">Portaal</namespace>
  <namespace key="101" case="first-letter">Portaalbesprekking</namespace>
  <namespace key="828" case="first-letter">Module</namespace>
  <namespace key="829" case="first-letter">Module talk</namespace>
  <namespace key="2300" case="first-letter">Gadget</namespace>
  <namespace key="2301" case="first-letter">Gadget talk</namespace>
</namespaces>

```

■ 7 Submitting large XML files

Integração de Sistemas

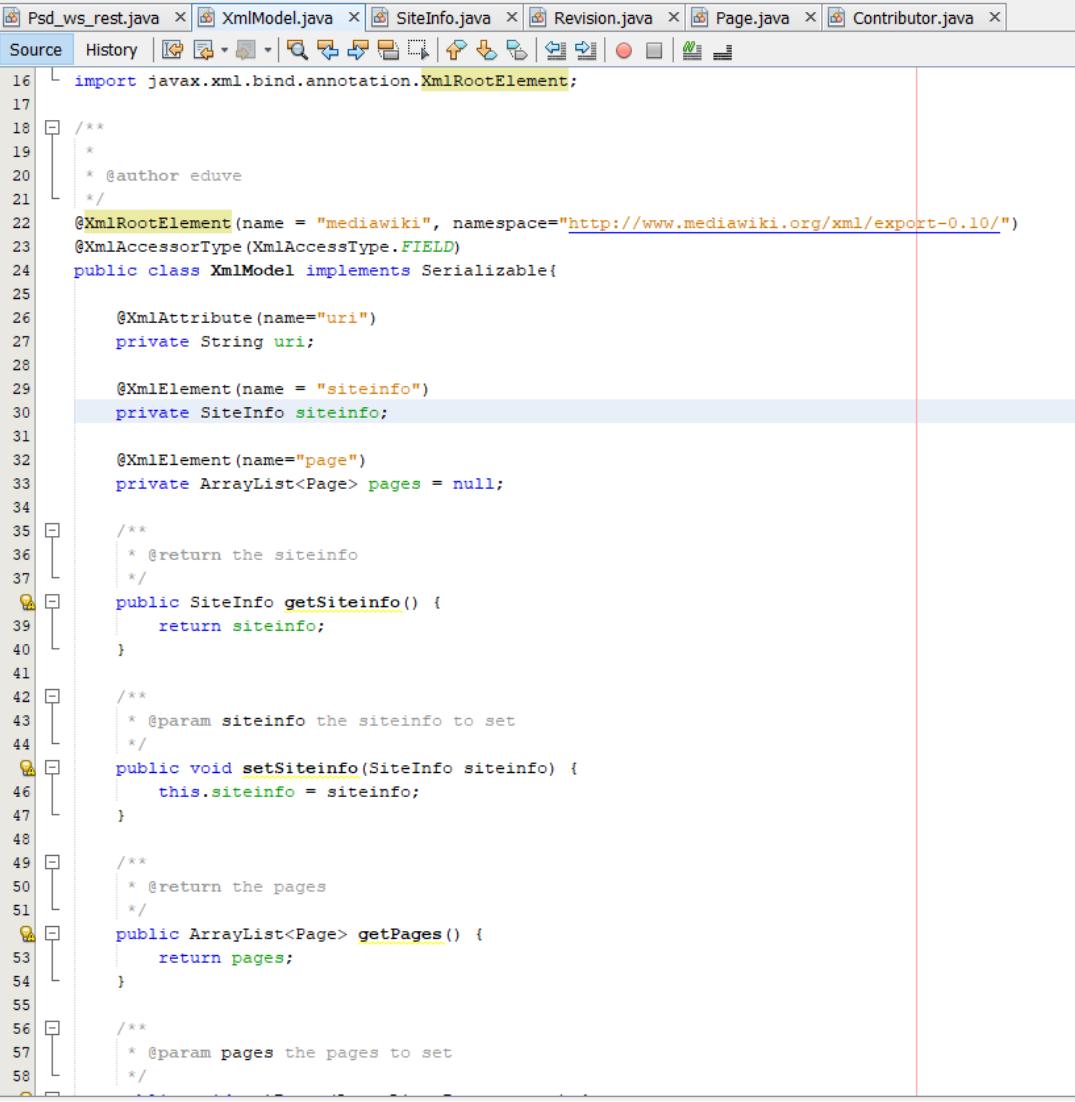
■ afwiki-20201101-pages-articles.xml file details



■ 7.1 REST webservice and parser JAXB - Java

Integração de Sistemas

- Java class XmlModel, to represent the structure of the xml file used.
- The rest of the classes can be examined by opening the project and analyzing the code.



```

16 import javax.xml.bind.annotation.XmlRootElement;
17
18 /**
19 * @author eduve
20 */
21
22 @XmlRootElement(name = "mediawiki", namespace="http://www.mediawiki.org/xml/export-0.10/")
23 @XmlAccessorType(XmlAccessType.FIELD)
24 public class XmlModel implements Serializable{
25
26     @XmlAttribute(name="uri")
27     private String uri;
28
29     @XmlElement(name = "siteinfo")
30     private SiteInfo siteinfo;
31
32     @XmlElement(name="page")
33     private ArrayList<Page> pages = null;
34
35 /**
36 * @return the siteinfo
37 */
38 public SiteInfo getSiteinfo() {
39     return siteinfo;
40 }
41
42 /**
43 * @param siteinfo the siteinfo to set
44 */
45 public void setSiteinfo(SiteInfo siteinfo) {
46     this.siteinfo = siteinfo;
47 }
48
49 /**
50 * @return the pages
51 */
52 public ArrayList<Page> getPages() {
53     return pages;
54 }
55
56 /**
57 * @param pages the pages to set
58 */

```

■ 7.1 REST webservice and parser JAXB - Java

Integração de Sistemas

- Attempt to load and parse the xml file with out of memory error.

IS_TP1 - NetBeans IDE 8.2 RC

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Psd_ws_rest.java

```

10 import java.io.File;
11 import java.io.IOException;
12 import java.lang.reflect.InvocationTargetException;
13 import java.util.ArrayList;
14 import javax.ws.rs.GET;
15 import javax.ws.rs.Path;
16 import javax.ws.rs.Produces;
17 import javax.ws.rs.core.MediaType;
18 import javax.xml.bind.JAXBContext;
19 import javax.xml.bind.JAXBException;
20 import javax.xml.bind.Unmarshaller;
21 import javax.xml.parsers.ParserConfigurationException;
22 import javax.xml.xpath.XPathExpressionException;
23 import org.xml.sax.SAXException;
24
25 @Path("/psd")
26 public class Psd_ws_rest {
27
28     @GET
29     @Produces(MediaType.APPLICATION_XML)
30     public ArrayList<String> getPsd() throws JAXBException, InvocationTargetException, XPathExpressionException, ParserConfigurationException, SAXException, IOException {
31
32         try {
33             JAXBContext jaxbContext;
34             File xmlFile = new File("C:\\Tutorial8\\FicheirosXML\\afwiki-20201101-pages-articles.xml");
35             jaxbContext = JAXBContext.newInstance(XmlModel.class);
36             Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();
37             XmlModel out = (XmlModel) jaxbUnmarshaller.unmarshal(xmlFile);
38
39         }
40     }
41 }
```

Output

```

IS_TP1 (run-deploy) × Java DB Database Process × Glassfish Server 4.1 ×
listening at address at http://LAPTOP-IUODEJBI:8080/IS_TP1_JAVA/Alunos_ws_soap.
Info: Webservice Endpoint deployed Books_ws_soap
listening at address at http://LAPTOP-IUODEJBI:8080/IS_TP1_JAVA/Books_ws_soap.
Info: Loading WS-TX Services. Please wait.
Info: visiting unvisited references
Info: Webservice Endpoint deployed ParticipantPortTypeImpl
listening at address at http://LAPTOP-IUODEJBI:8080/_wstx-services/ParticipantPortType.
Info: Webservice Endpoint deployed RegistrationPortTypeRPCPortImpl
listening at address at http://LAPTOP-IUODEJBI:8080/_wstx-services/RegistrationPortTypeRPC11.
Info: Webservice Endpoint deployed CoordinatorPortTypeImpl
listening at address at http://LAPTOP-IUODEJBI:8080/_wstx-services/CoordinatorPortType.
Info: Webservice Endpoint deployed RegistrationRequesterPortTypePortImpl
listening at address at http://LAPTOP-IUODEJBI:8080/_wstx-services/RegistrationRequesterPortTypeImpl.
Info: Webservice Endpoint deployed CoordinatorPortTypePortImpl
```

HTTP Server Monitor Search Results Navigator 34:101 INS

■ 7.2 SOAP webservice and parser DOM – PHP

Integração de Sistemas

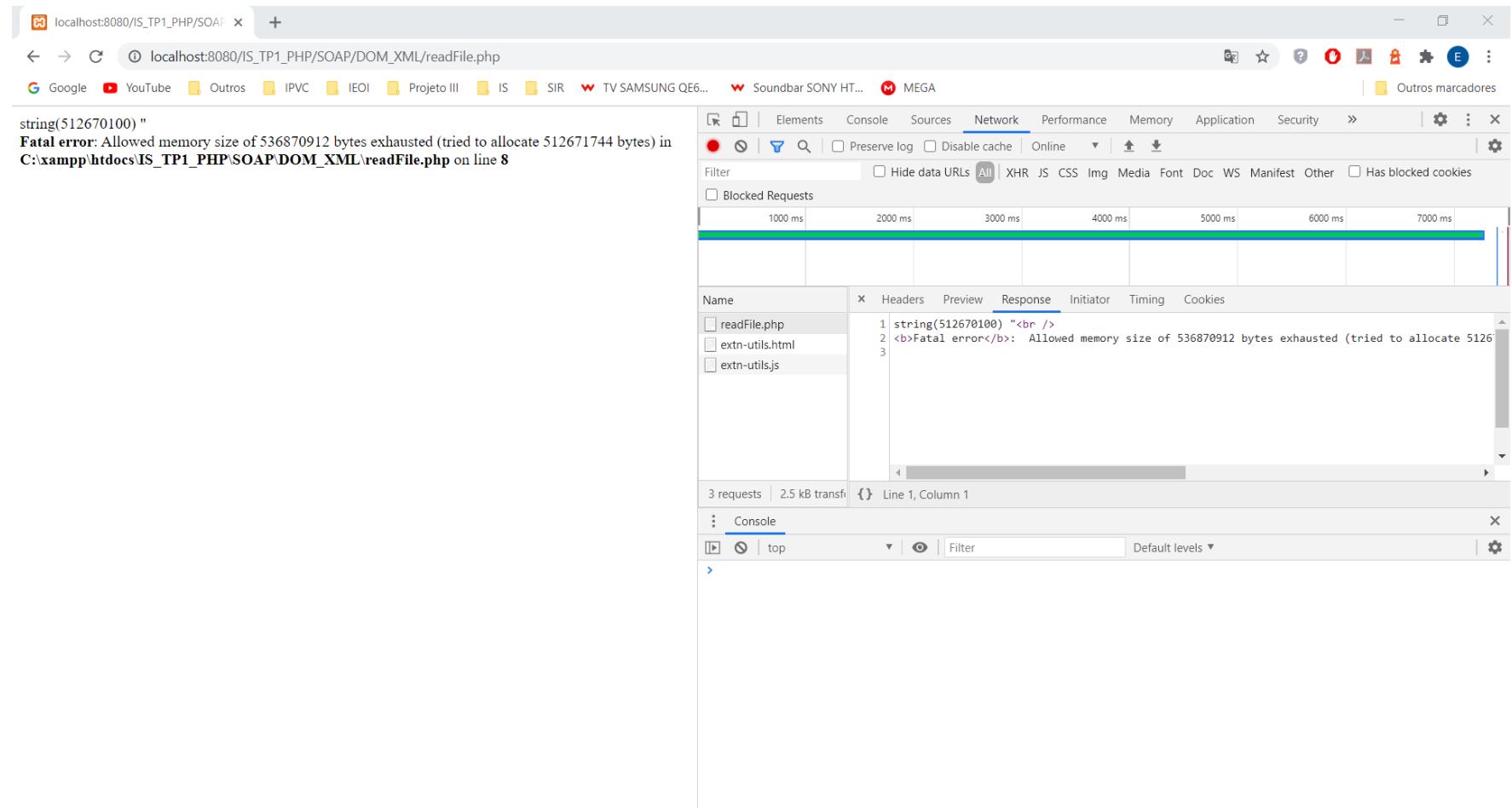
- afwiki-20201101-pages-articles.xml imported in the readFile.php file

```
readFile.php X
SOAP > DOM_XML > readFile.php > ...
1 <?php
2     getDataFile();
3     function getDataFile() {
4         try {
5             $dom = new domDocument;
6
7             $dom->load("files/afwiki-20201101-pages-articles.xml");
8
9             echo $dom->saveXML();
10        } catch(Exception $e) {
11            echo $e->getMessage();
12        }
}
```

■ 7.2 SOAP webservice and parser DOM – PHP

Integração de Sistemas

- Attempt to parse and load the xml file with out of memory error.



A screenshot of a web browser window showing an error message. The URL in the address bar is `localhost:8080/IS_TP1_PHP/SOAP/`. The error message displayed is:

```
string(512670100) "
Fatal error: Allowed memory size of 536870912 bytes exhausted (tried to allocate 512671744 bytes) in
C:\xampp\htdocs\IS_TP1_PHP\SOAP\DOM_XML\readFile.php on line 8
```

The browser's developer tools Network tab is open, showing a list of requests. One request, `readFile.php`, is selected, and its response is shown in the Response panel:

```
1 string(512670100) "<br />
2 <b>Fatal error</b>: Allowed memory size of 536870912 bytes exhausted (tried to allocate 512671744 bytes) in
3 C:\xampp\htdocs\IS_TP1_PHP\SOAP\DOM_XML\readFile.php on line 8"
```

The Network tab also shows a timeline of requests and a list of blocked requests.

■ 7.3 Problems and approaches to process large files

■ Main problems in processing and parsing a large files [8]:

- The first and more obvious problem is the lack of memory on the server side, which results in a fatal error and the immediate termination of the process during the processing stage of the referenced file. This error was the one verified in the attempts registered in the points 7.2 and 7.3, where both the glassfish server and apache server could not parse and process the file due to the allowed memory size for these servers.
- Being the file large, the transfer could be interrupted because of the server timeout, not receiving the file consequently.
- The server includes a limit to the files uploaded, if the file's size is over the imposed limit, the connection is severed, and the file is not uploaded.

■ 7.3 Problems and approaches to process large files

Integração de Sistemas

■ Ways to handle the upload and transfer of large files [9]:

- The most common way to be able to upload and transfer large files, is breaking them into several pieces and processing them one at a time. This process has one big advantage, and it is that if eventually one of the chunks arrive with errors the process can restart from where it was without losing all the progress done until that point.
- A second way could involve asynchronous uploads, where the file is uploaded in the background allowing the software to continue functioning, reducing the risks of a timeout or memory caps.
- Uploading the file directly to a cloud service and accessing it from there could be an option but it has drawbacks, like the long upload time if it needs to be processed in several locations before it reaches the cloud.

■ 7.3 Problems and approaches to process large files

Integração de Sistemas

- What would need to be changed in order to receive a large file in the Java or PHP examples:
 - In Java, a solution could be the use of the *AsyncXMLInputFactory*, in order to parse the incoming XML file in an asynchronous way, parsing it in the background while the service continues to run. This also allows to pull parts of the file, parsing it piece by piece, reducing the risk of hitting the server's memory and timeout caps [10].
 - In PHP, instead of using an XML parser like DOM, which loads all the contents of the XML file into memory, using the XMLReader and the functions it provides, it would be possible to pull one or more elements at a time from the file, loading them into memory, not overloading the server and using a very reduced memory size and with almost no risk of timeouts [11].

■ 8 SOAP and REST web services implementation in Python and JavaScript

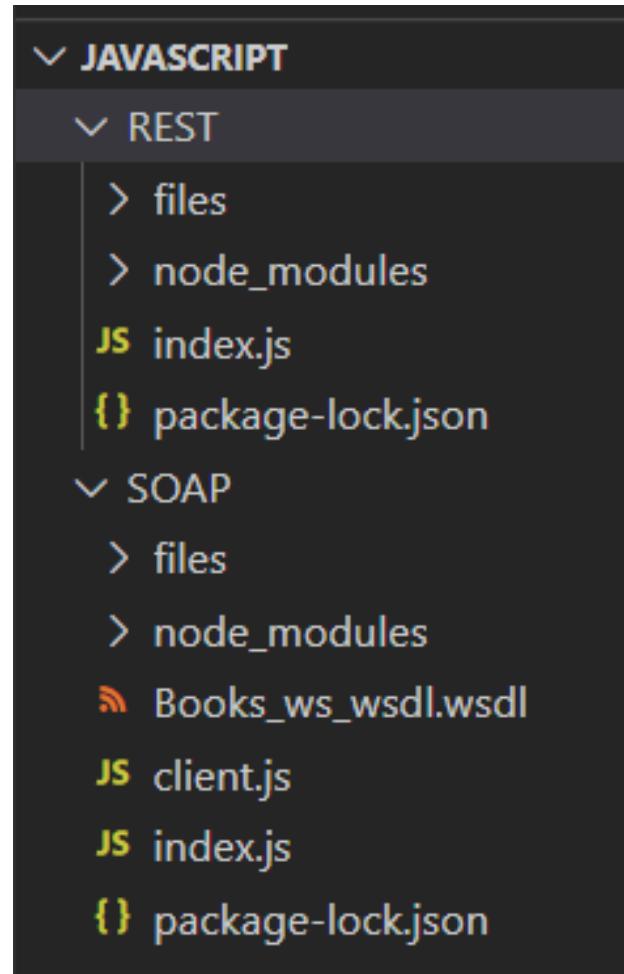
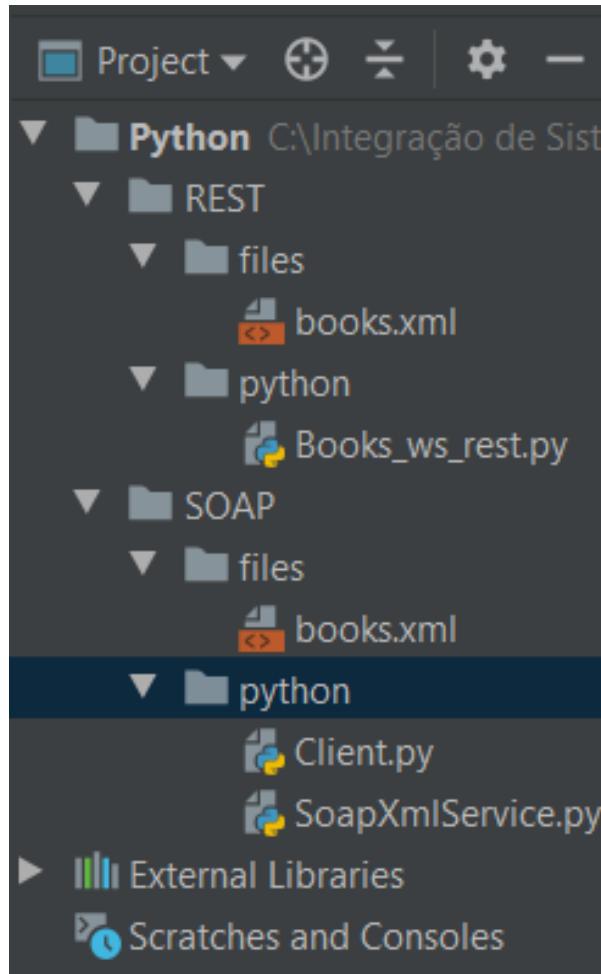
Integração de Sistemas

- This section shows the implementation done of REST and SOAP web services in Python and in JavaScript, using the file books.xml and XML parsers to de-serialize the contents of this file and return the data obtained to the client. Additionally these web services will also use XPATH to filter through the contents of the XML file and return only the results matching a certain expression.

- To achieve this goal, it was created a project for the web services in Python using the PyCharm IDE and for the JavaScript a project with Visual Studio Code. The final project's structure is the one presented in the following slide.

■ 8 SOAP and REST web services implementation in Python and JavaScript

Integração de Sistemas



■ 8.1 Python – REST web service

Integração de Sistemas

- All the REST web services in Python were created using the library *Flask*, which allows the creation of a server where it is possible to register several endpoints to be by the client, which in this case is the browser. Additionally it was also used the *lxml* and the *xml.dom* libraries more specifically in the parsing and reading of the xml file.
- In this first implementation there are two endpoints available, in which one of them execute the parse of the file books.xml, whereas the second one executes the parts of the file and uses XPath expressions to filter through its contents, returning only the ones who match the referred expression. Accessing the root endpoint, two web links are shown and if clicked they call correspondent method on the web service.
- In the following slides the pictures are accompanied by a description of what they are presenting

■ 8.1 Python – REST web service

Integração de Sistemas

- Imports, root endpoint (at '/'), `getBooks` function and the correspondent endpoint to parse the `books.xml` file, for argument type html.

```
Books_ws_rest.py x
1  from xml.dom import minidom
2  from flask import Flask, request
3  from lxml import etree
4  from werkzeug.wrappers import Response
5  app = Flask(__name__)
6
7
8  @app.route('/')
9  def index():
10      html = "<h3>Esta aplicação contém os seguintes métodos: </h3>"
11      html = html + "<a href='/getBooks/books'>XML Parse - Ficheiro books.xml</a><p>Resposta em XML e HTML</p><br>"
12      html = html + "<a href='/getBooks/booksXpath'>XML Parse e XPATH - Ficheiro books.xml</a>"
13      return html
14
15  @app.route('/getBooks/books', methods=['GET'])
16  def getBooks():
17      typeRequest = request.args.get('type')
18      if typeRequest.upper() == "HTML":
19          xmldoc = minidom.parse('../files/books.xml')
20          itemlist = xmldoc.getElementsByTagName('book')
21          data = ""
22          i = 0
23          for i in range(0, len(itemlist)):
24              node = itemlist.item(i)
25              autor = node.getElementsByTagName('author')[0].childNodes[0].nodeValue
26              titulo = node.getElementsByTagName('title')[0].childNodes[0].nodeValue
27              genero = node.getElementsByTagName('genre')[0].childNodes[0].nodeValue
28              preco = node.getElementsByTagName('price')[0].childNodes[0].nodeValue
29              data_publicacao = node.getElementsByTagName('publish_date')[0].childNodes[0].nodeValue
30              descricao = node.getElementsByTagName('description')[0].childNodes[0].nodeValue
31              data = data + "<h5>Autor: " + autor + "</h5>"
32              data = data + "<h5>Titolo: " + titulo + "</h5>"
```

■ 8.1 Python – REST web service

Integração de Sistemas

```

Books_ws_rest.py ×

33     data = data + "<h5>Genero: " + genero + "<h5>"
34     data = data + "<h5>Preco: " + preco + "<h5>"
35     data = data + "<h5>Data de Publicação: " + data_publicacao + "<h5>"
36     data = data + "<h5>Descrição: " + descricao + "<h5>"
37     data = data + "<br>"
38     return data
39 if typeRequest.upper() == "XML":
40     root = etree.parse('../files/books.xml').getroot()
41     xml = etree.tostring(root, pretty_print=True)
42     return Response(xml, mimetype='text/xml')
43
44 @app.route('/getBooks/booksXPath', methods=['GET'])
45 def getBooksXPath():
46     root = etree.parse('../files/books.xml')
47     books = root.xpath('/catalog//book[price<7.00 and price>5.00]')
48     data = "<h1>Resultados da pesquisa XPATH por: </h1><h2>price < 7.00 and price > 5.00</h2><br>"
49     i = 0
50     for i in range(0, len(books)):
51         autor = books[i][0].text
52         titulo = books[i][1].text
53         genero = books[i][2].text
54         preco = books[i][3].text
55         data_publicacao = books[i][4].text
56         descricao = books[i][5].text
57         data = data + "<h5>Autor: " + autor + "<h5>"
58         data = data + "<h5>Título: " + titulo + "<h5>"
59         data = data + "<h5>Genero: " + genero + "<h5>"
60         data = data + "<h5>Preço: " + preco + "<h5>"
61         data = data + "<h5>Data de Publicação: " + data_publicacao + "<h5>"
62         data = data + "<h5>Descrição: " + descricao + "<h5>"
63         data = data + "<br>"
64     return data

```

■ Return of the constructed string

with html, xml type request and

the function *getBooksXPath* to

filter the results using an Xpath

expression.

■ Main function to start the service

```

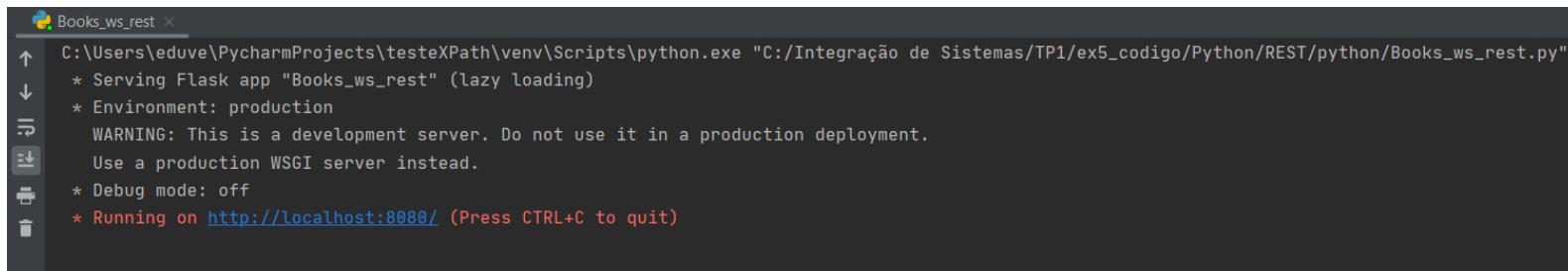
65
66 ► if __name__ == "__main__":
67     app.run(host='localhost', port=8080)
68

```

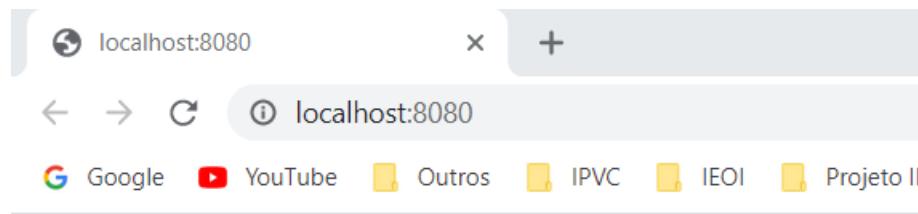
■ 8.1 Python – REST web service

Integração de Sistemas

- Web service running on localhost, port 8080, and accessing the root endpoint through the browser



```
Books_ws_rest x
C:\Users\eduve\PycharmProjects\testeXPath\venv\Scripts\python.exe "C:/Integração de Sistemas/TP1/ex5_codigo/Python/REST/python/Books_ws_rest.py"
* Serving Flask app "Books_ws_rest" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://localhost:8080/ (Press CTRL+C to quit)
```



Esta aplicação contém os seguintes métodos:

XML Parse - Ficheiro books.xml

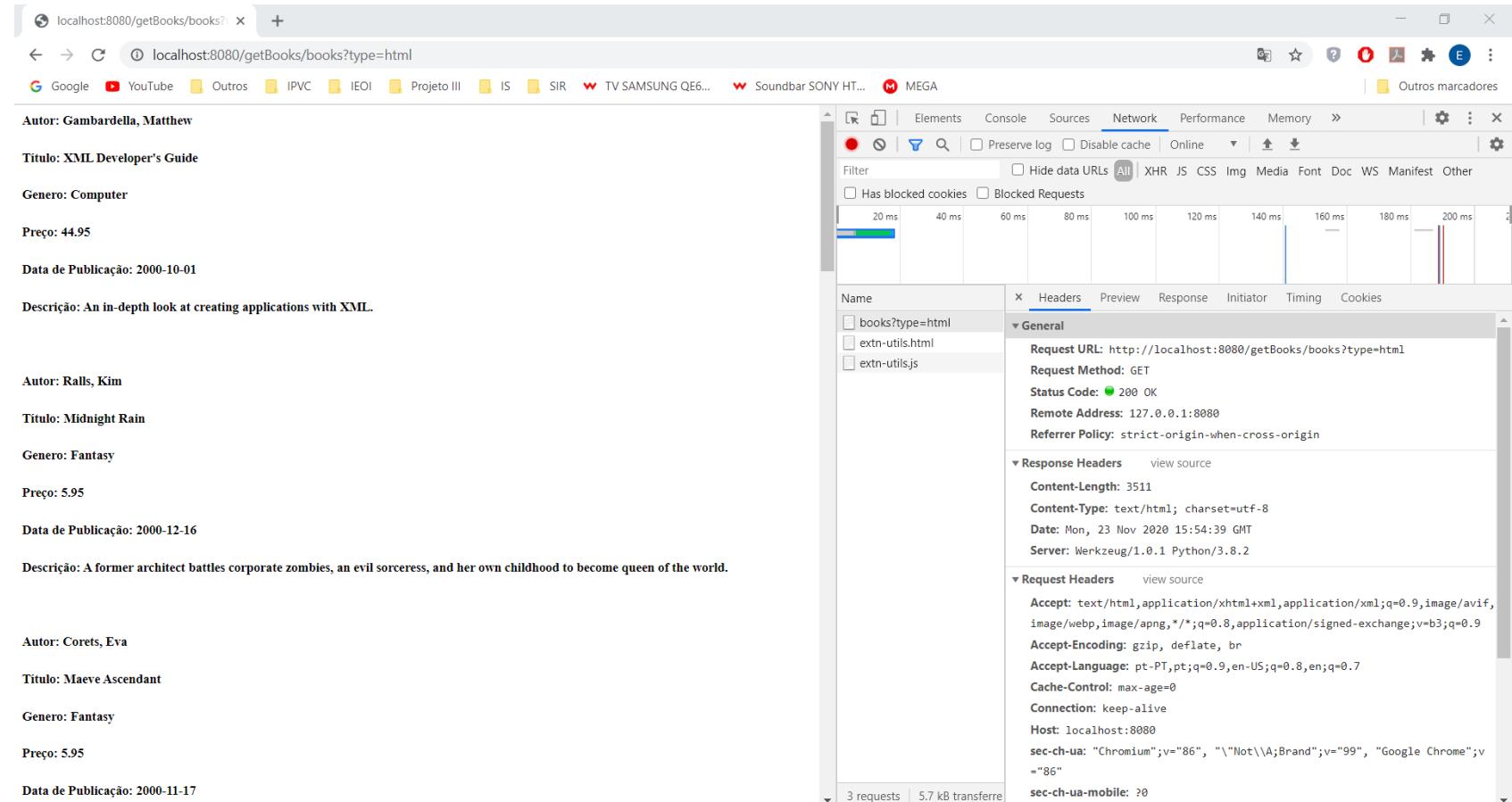
Resposta em XML e HTML

XML Parse e XPATH - Ficheiro books.xml

■ 8.1 Python – REST web service

Integração de Sistemas

- Calling the `getBooks/books` endpoint specifying the type of output as html



Autor: Gambardella, Matthew
 Título: XML Developer's Guide
 Gênero: Computer
 Preço: 44.95
 Data de Publicação: 2000-10-01
 Descrição: An in-depth look at creating applications with XML.

Autor: Ralls, Kim
 Título: Midnight Rain
 Gênero: Fantasy
 Preço: 5.95
 Data de Publicação: 2000-12-16
 Descrição: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Autor: Corets, Eva
 Título: Maeve Ascendant
 Gênero: Fantasy
 Preço: 5.95
 Data de Publicação: 2000-11-17

Name	Headers	Preview	Response	Initiator	Timing	Cookies
books?type=html						
extn-utils.html						
extn-utils.js						

Request URL: http://localhost:8080/getBooks/books?type=html
 Request Method: GET
 Status Code: 200 OK
 Remote Address: 127.0.0.1:8080
 Referrer Policy: strict-origin-when-cross-origin

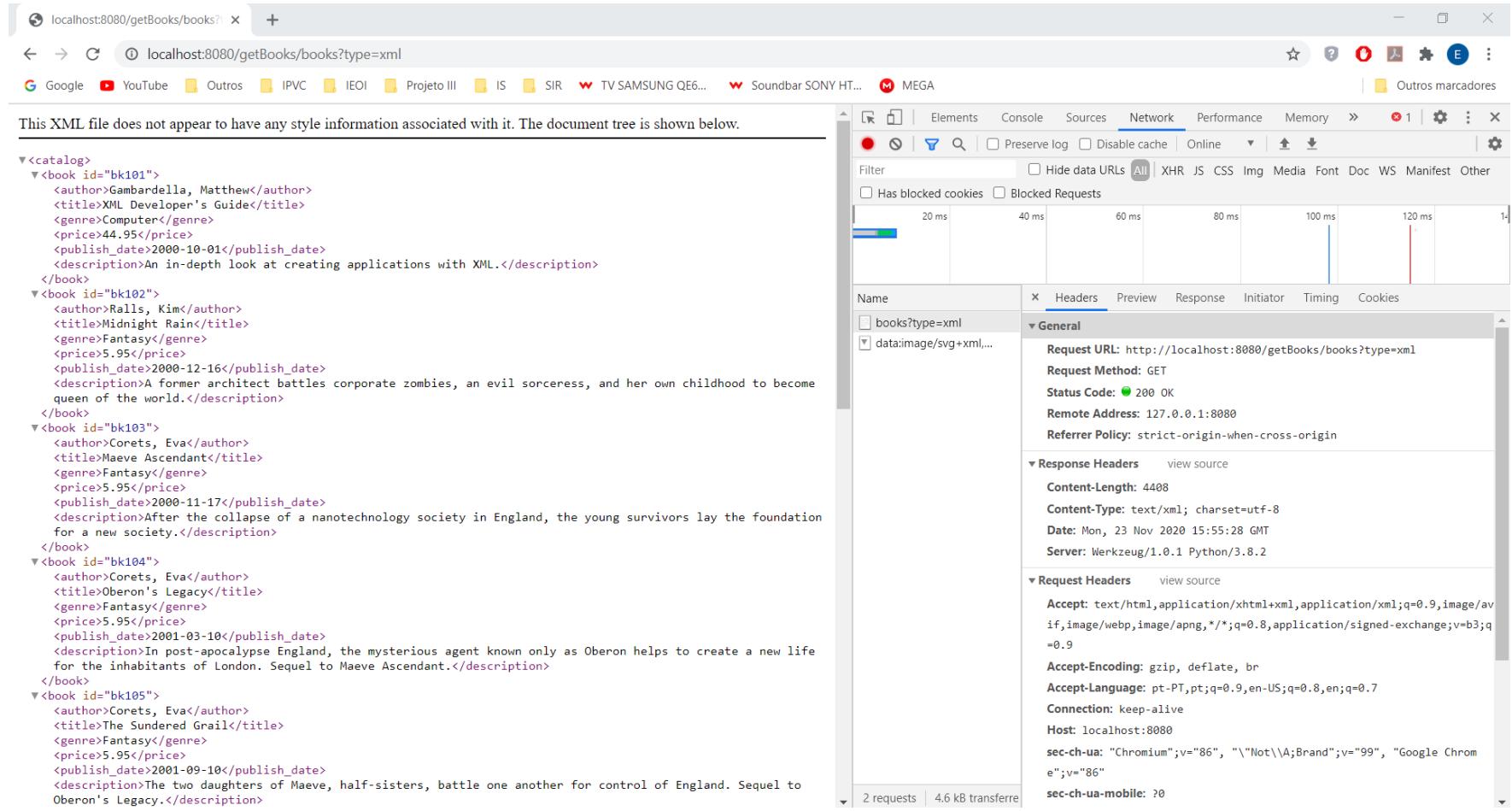
Content-Length: 3511
 Content-Type: text/html; charset=utf-8
 Date: Mon, 23 Nov 2020 15:54:39 GMT
 Server: Werkzeug/1.0.1 Python/3.8.2

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
 Accept-Encoding: gzip, deflate, br
 Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
 Cache-Control: max-age=0
 Connection: keep-alive
 Host: localhost:8080
 sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome";v="86"
 sec-ch-ua-mobile: ?0

■ 8.1 Python – REST web service

Integração de Sistemas

■ Calling the *getBooks/books* endpoint specifying the type of output as xml



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.</description>
  </book>
  <book id="bk103">
    <author>Corets, Eva</author>
    <title>Maeve Ascendant</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-11-17</publish_date>
    <description>After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.</description>
  </book>
  <book id="bk104">
    <author>Corets, Eva</author>
    <title>Oberon's Legacy</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-03-10</publish_date>
    <description>In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.</description>
  </book>
  <book id="bk105">
    <author>Corets, Eva</author>
    <title>The Sundered Grail</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2001-09-10</publish_date>
    <description>The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.</description>
  </book>
```

■ 8.1 Python – REST web service

Integração de Sistemas

- Calling the `getBooks/booksXpath` endpoint and the result obtained

Resultados da pesquisa XPATH por:

price < 7.00 and price > 5.00

Autor: Ralls, Kim

Titulo: Midnight Rain

Genero: Fantasy

Preço: 5.95

Data de Publicação: 2000-12-16

Descrição: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Autor: Corets, Eva

Titulo: Maeve Ascendant

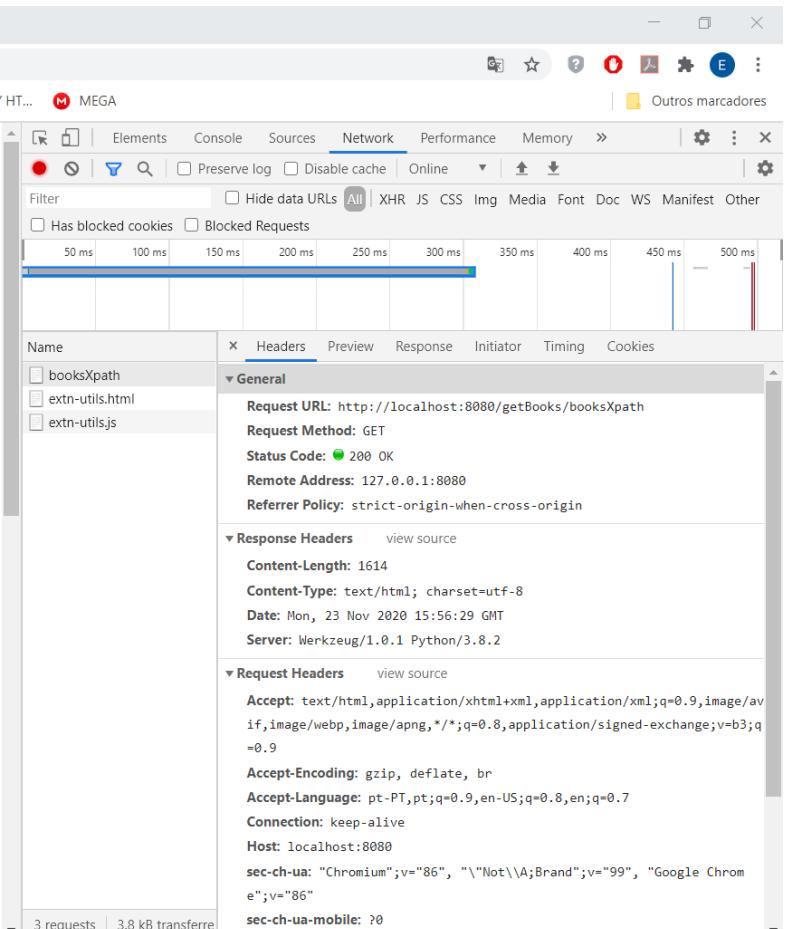
Genero: Fantasy

Preço: 5.95

Data de Publicação: 2000-11-17

Descrição: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.

Autor: Corets, Eva



The screenshot shows the Network tab of the Chrome DevTools Network panel. It displays a single request for the URL `http://localhost:8080/getBooks/booksXpath`. The request was made via GET and received a 200 OK status code. The response body contains the XML data from the REST API endpoint, which is identical to the content shown on the left side of the screen. The total transfer size is 3.8 kB.

■ 8.2 Python – SOAP web service

Integração de Sistemas

- As far as soap web services in Python go, in this practical work, they were implemented using the Spyne framework, which allows to create a server, registering created methods and mapping the web service output and input types, which in this case are both SOAP11.

- As it was done for the rest web service this web service has two endpoints available, where one of them executes the parse of the XML file, and the other one uses an XPath expression obtaining only the books which price is between 7.00 and 5.00.

■ 8.2 Python – SOAP web service

Integração de Sistemas

- Necessary imports, *books_parse* function, the necessary code using minidom to parse the file and obtain an element list and the construction of the html string to send back to the client.

```
SoapXmlService.py ×
1  from spyne import Application, rpc, ServiceBase, Unicode
2  from spyne.protocol.soap import Soap11
3  from spyne.server.wsgi import WsgiApplication
4  from xml.dom import minidom
5  from lxml import etree
6
7
8  class SoapXmlService(ServiceBase):
9      @rpc(_returns=Unicode)
10     def books_parse(ctx):
11         xmldoc = minidom.parse('../files/books.xml')
12         itemlist = xmldoc.getElementsByTagName('book')
13         data = ""
14         i = 0
15         for i in range(0, len(itemlist)):
16             node = itemlist.item(i)
17             autor = node.getElementsByTagName('author')[0].childNodes[0].nodeValue
18             titulo = node.getElementsByTagName('title')[0].childNodes[0].nodeValue
19             genero = node.getElementsByTagName('genre')[0].childNodes[0].nodeValue
20             preco = node.getElementsByTagName('price')[0].childNodes[0].nodeValue
21             data_publicacao = node.getElementsByTagName('publish_date')[0].childNodes[0].nodeValue
22             descricao = node.getElementsByTagName('description')[0].childNodes[0].nodeValue
23             data = data + "<h5>Autor: " + autor + "<h5>"
24             data = data + "<h5>Título: " + titulo + "<h5>"
25             data = data + "<h5>Genero: " + genero + "<h5>"
26             data = data + "<h5>Preco: " + preco + "<h5>"
27             data = data + "<h5>Data de Publicação: " + data_publicacao + "<h5>"
28             data = data + "<h5>Descrição: " + descricao + "<h5>"
29             data = data + "<br>"
30
31     return data
```

■ 8.2 Python – SOAP web service

Integração de Sistemas

- Second method, *books_Xpath*, using etree from the lxml library to filter through all the books in the books.xml file and return only the list of elements who verify the expression, building the html

```
SoapXmlService.py ×
31
32     @rpc(_returns=Unicode)
33     def books_Xpath(ctx):
34         root = etree.parse('../files/books.xml')
35         books = root.xpath('/catalog//book[price<7.00 and price>5.00]')
36         data = "<h1>Resultados da pesquisa XPATH por: </h1><h2>price < 7.00 and price > 5.00</h2><br>"
37         i = 0
38         for i in range(0, len(books)):
39             autor = books[i][0].text
40             titulo = books[i][1].text
41             genero = books[i][2].text
42             preco = books[i][3].text
43             data_publicacao = books[i][4].text
44             descricao = books[i][5].text
45             data = data + "<h5>Autor: " + autor + "<h5>"
46             data = data + "<h5>Título: " + titulo + "<h5>"
47             data = data + "<h5>Genero: " + genero + "<h5>"
48             data = data + "<h5>Preco: " + preco + "<h5>"
49             data = data + "<h5>Data de Publicação: " + data_publicacao + "<h5>"
50             data = data + "<h5>Descrição: " + descricao + "<h5>"
51             data = data + "<br>"
52
53
54
55         application = Application([SoapXmlService],
56                                 tns='soap.books.parse',
57                                 in_protocol=Soap11(validation='lxml'),
58                                 out_protocol=Soap11()
59

```

to send to the client.

- Additionally the web service is registered in the application, referring a *tns*, the *in_protocol* and the *out_protocol*.

■ 8.2 Python – SOAP web service

Integração de Sistemas

- Main function where the server is created and started in localhost in the port 8000, in this case.

```
60 ►  if __name__ == '__main__':
61      from wsgiref.simple_server import make_server
62
63      wsgi_app = WsgiApplication(application)
64      server = make_server('localhost', 8000, wsgi_app)
65      server.serve_forever()
66
```

■ 8.2 Python – SOAP web service

Integração de Sistemas

■ For the client, it was created a flask server with methods that can be called from the browser and create using the zeep library, a client, which calls the created webservice and the respective method.

```

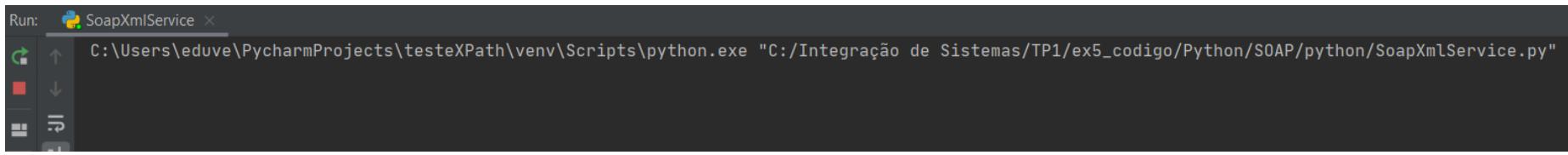
1  from zeep import Client
2  from flask import Flask
3  app = Flask(__name__)
4
5
6  @app.route('/')
7  def index():
8      html = "<h3>Servidor Flask para teste do Web service em SOAP: </h3>"
9      html = html + "<a href='/soap/booksParse'>XML Parse - Ficheiro books.xml</a><br>"
10     html = html + "<a href='/soap/Xpath'>XPATH - Ficheiro books.xml</a>"
11     return html
12
13 @app.route('/soap/booksParse', methods=['GET'])
14 def booksParseSOAP():
15     client = Client(wsdl='http://localhost:8000/soap.books.parse?wsdl')
16     data = client.service.books_parse()
17     return data
18
19 @app.route('/soap/Xpath', methods=['GET'])
20 def booksXPath():
21     client = Client(wsdl='http://localhost:8000/soap.books.parse?wsdl')
22     data = client.service.books_Xpath()
23     return data
24
25 if __name__ == "__main__":
26     app.run(host='localhost', port=8080)
27

```

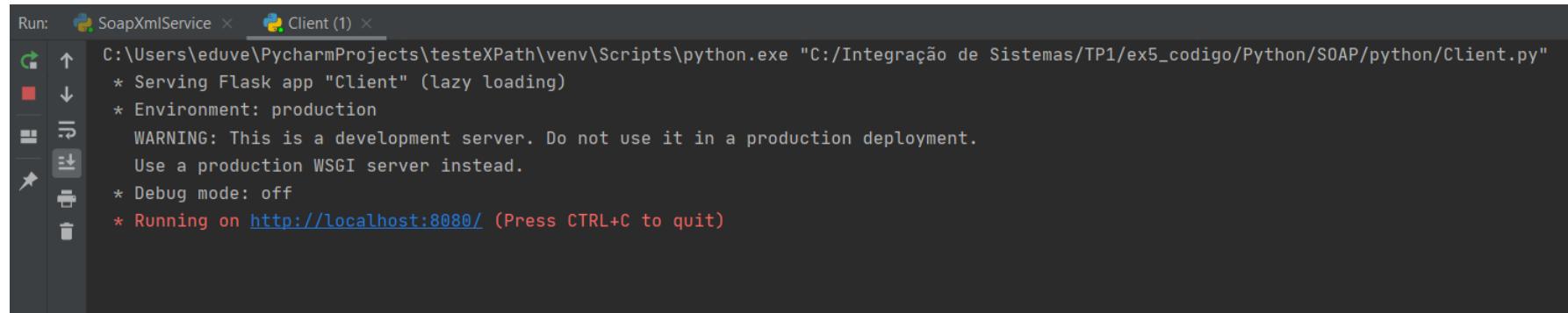
■ 8.2 Python – SOAP web service

Integração de Sistemas

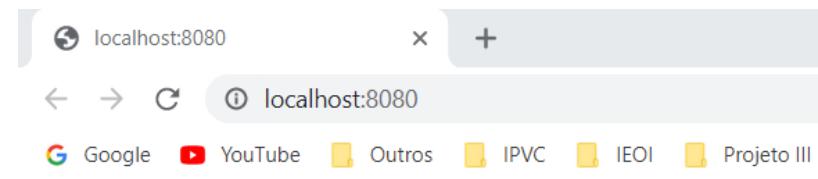
- From top to bottom, the first image is the web service running, the second one demonstrates the client running, and the third one presents the root endpoint in the client server, used to present the data.



Run: SoapXmlService x
C:\Users\eduve\PycharmProjects\testeXPath\venv\Scripts\python.exe "C:/Integração de Sistemas/TP1/ex5_codigo/Python/SOAP/python/SoapXmlService.py"



Run: SoapXmlService x Client (1) x
C:\Users\eduve\PycharmProjects\testeXPath\venv\Scripts\python.exe "C:/Integração de Sistemas/TP1/ex5_codigo/Python/SOAP/python/Client.py"
* Serving Flask app "Client" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on <http://localhost:8080/> (Press CTRL+C to quit)



Servidor Flask para teste do Web service em SOAP:

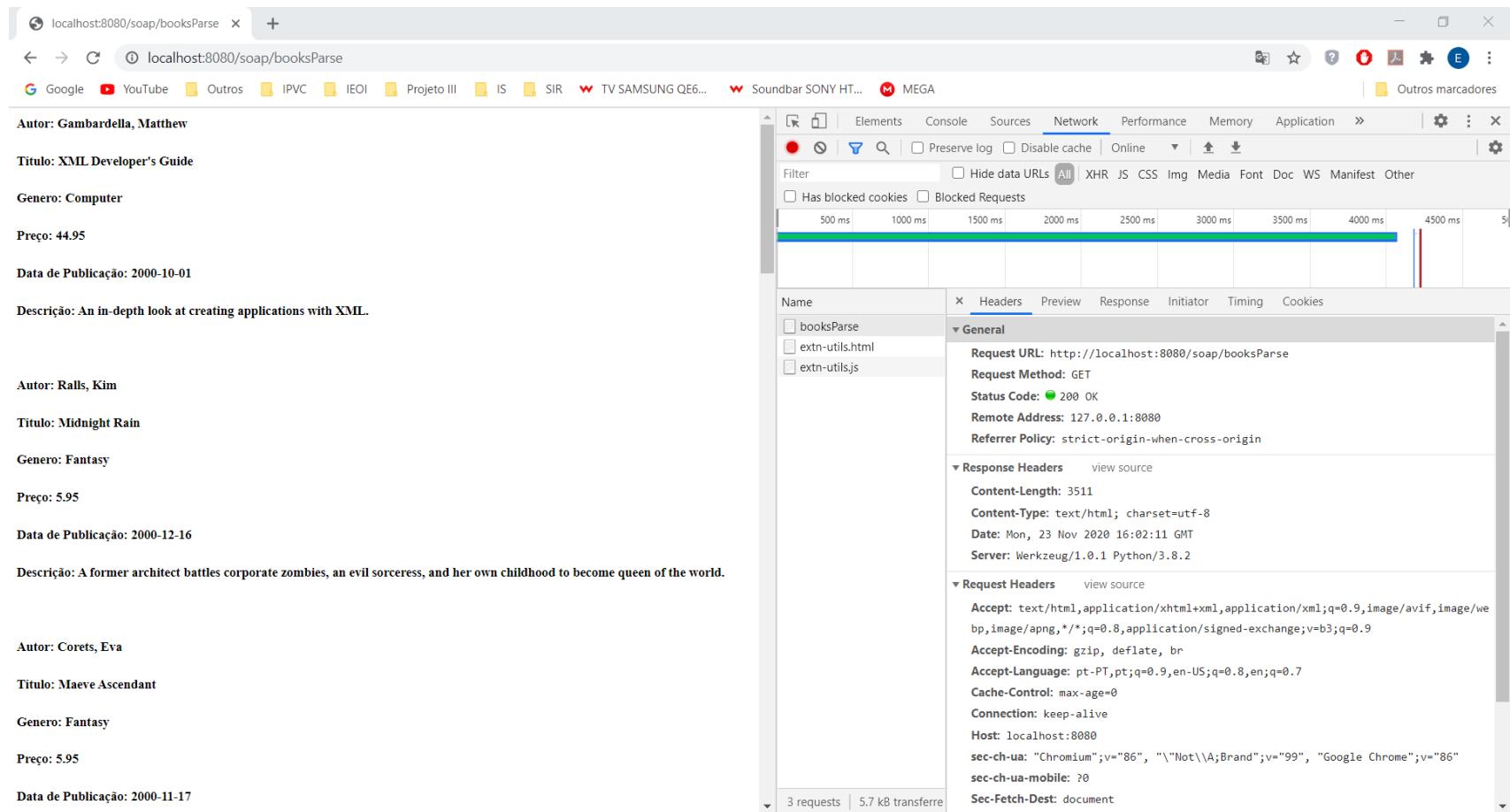
[XML Parse - Ficheiro books.xml](#)

[XPATH - Ficheiro books.xml](#)

■ 8.2 Python – SOAP web service

Integração de Sistemas

- Result of calling the *books_parse* method on the web service, with the data presented on the client side.



Autor: Gambardella, Matthew

Título: XML Developer's Guide

Gênero: Computer

Preço: 44.95

Data de Publicação: 2000-10-01

Descrição: An in-depth look at creating applications with XML.

Autor: Ralls, Kim

Título: Midnight Rain

Gênero: Fantasy

Preço: 5.95

Data de Publicação: 2000-12-16

Descrição: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Autor: Corets, Eva

Título: Maeve Ascendant

Gênero: Fantasy

Preço: 5.95

Data de Publicação: 2000-11-17

Name	Headers	Preview	Response	Initiator	Timing	Cookies
booksParse						
extn-utils.html						
extn-utils.js						

General

- Request URL: http://localhost:8080/soap/booksParse
- Request Method: GET
- Status Code: 200 OK
- Remote Address: 127.0.0.1:8080
- Referrer Policy: strict-origin-when-cross-origin

Response Headers

- Content-Length: 3511
- Content-Type: text/html; charset=utf-8
- Date: Mon, 23 Nov 2020 16:02:11 GMT
- Server: Werkzeug/1.0.1 Python/3.8.2

Request Headers

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
- Accept-Encoding: gzip, deflate, br
- Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
- Cache-Control: max-age=0
- Connection: keep-alive
- Host: localhost:8080
- sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome";v="86"
- sec-ch-ua-mobile: ?0
- Sec-Fetch-Dest: document

■ 8.2 Python – SOAP web service

Integração de Sistemas

- Result of calling the *books_Xpath* method and the result presented in the browser.

Resultados da pesquisa XPATH por:

price < 7.00 and price > 5.00

Autor: Ralls, Kim

Titulo: Midnight Rain

Genero: Fantasy

Preço: 5.95

Data de Publicação: 2000-12-16

Descrição: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Autor: Corets, Eva

Titulo: Maeve Ascendant

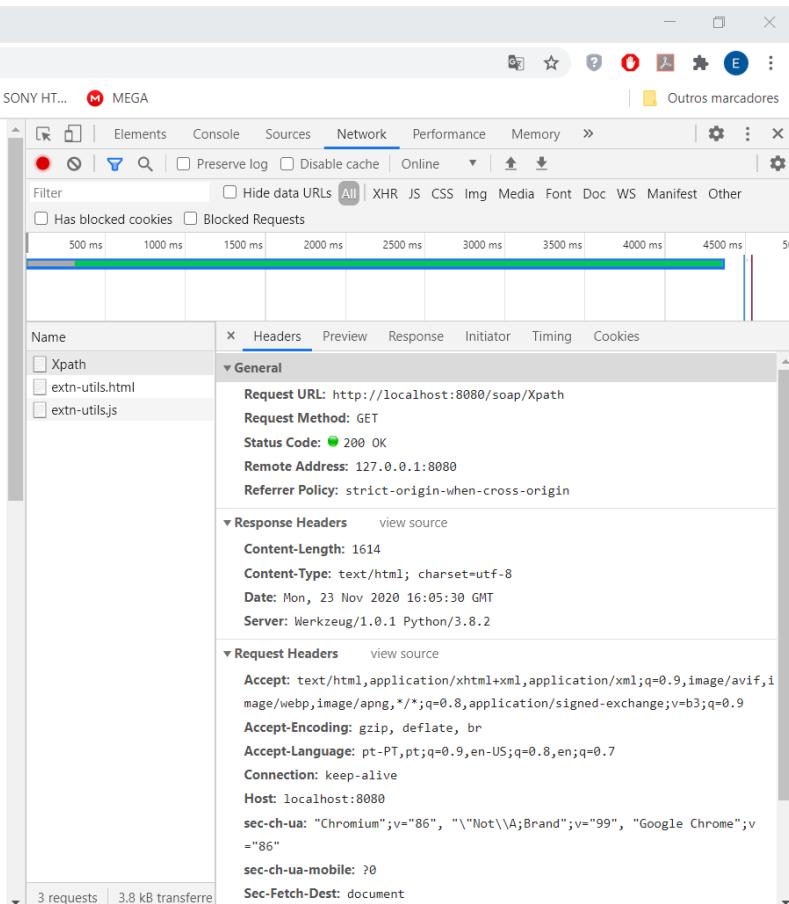
Genero: Fantasy

Preço: 5.95

Data de Publicação: 2000-11-17

Descrição: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.

Autor: Corets, Eva



Request URL: http://localhost:8080/soap/Xpath

Request Method: GET

Status Code: 200 OK

Remote Address: 127.0.0.1:8080

Referrer Policy: strict-origin-when-cross-origin

Content-Length: 1614

Content-Type: text/html; charset=utf-8

Date: Mon, 23 Nov 2020 16:05:30 GMT

Server: Werkzeug/1.0.1 Python/3.8.2

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Accept-Encoding: gzip, deflate, br

Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7

Connection: keep-alive

Host: localhost:8080

sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome";v="86"

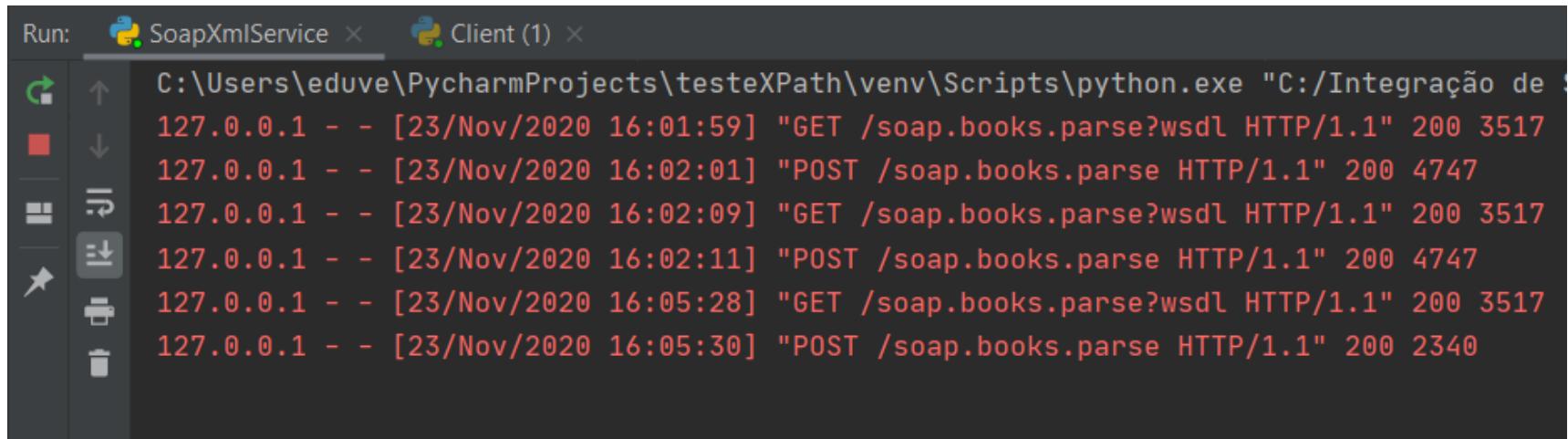
sec-ch-ua-mobile: ?0

Sec-Fetch-Dest: document

■ 8.2 Python – SOAP web service

Integração de Sistemas

- Registered calls to the web service.



The screenshot shows the PyCharm interface with the 'Run' tab open. The 'SoapXmlService' run configuration is selected. The terminal window displays a list of registered calls to the web service:

```
C:\Users\eduve\PycharmProjects\testeXPath\venv\Scripts\python.exe "C:/Integração de Sistemas/SoapXmlService.py"
127.0.0.1 - - [23/Nov/2020 16:01:59] "GET /soap.books.parse?wsdl HTTP/1.1" 200 3517
127.0.0.1 - - [23/Nov/2020 16:02:01] "POST /soap.books.parse HTTP/1.1" 200 4747
127.0.0.1 - - [23/Nov/2020 16:02:09] "GET /soap.books.parse?wsdl HTTP/1.1" 200 3517
127.0.0.1 - - [23/Nov/2020 16:02:11] "POST /soap.books.parse HTTP/1.1" 200 4747
127.0.0.1 - - [23/Nov/2020 16:05:28] "GET /soap.books.parse?wsdl HTTP/1.1" 200 3517
127.0.0.1 - - [23/Nov/2020 16:05:30] "POST /soap.books.parse HTTP/1.1" 200 2340
```

■ 8.3 JavaScript – REST web service

Integração de Sistemas

- For the implementation of a REST web service in JavaScript, it was used Node Js, in order to create a server where all the created methods to access, parse and filer the contents of the books.xml file will be registered.
- The first endpoint, obtains the arguments from the URL and checks if the argument *file* is equal to *books*, and if it is, then the file to execute the parsing is books.xml. In this case the file is parsed, and the contents returned are in a form of a string.
- The second endpoint verifies the arguments in the same way as the first one, but it only extracts the nodes from the XML file that match the XPath expression, and returning html to the client with the results.

■ 8.3 JavaScript – REST web service

Integração de Sistemas

■ Necessary imports, first endpoint registered to execute the parse of the books.xml file, constructing the string to send the data back to the client.

```
JS index.js  X
REST > JS index.js > app.get('/booksXpath') callback
1  var express = require('express');
2  var app = express();
3  var fs = require("fs");
4  var parser = require('xml2json-light');
5  var xpath = require('xpath')
6  var dom = require('xmldom').DOMParser
7  const http = require('http');
8  const url = require('url');

9
10 app.get('/booksParse', function (req, res) {
11   const queryObject = url.parse(req.url, true).query;
12   if (queryObject["file"] == "books") {
13     fs.readFile("files/books.xml", 'utf8', function (err, data) {
14       var json = parser.xml2json(data);
15       var i = 0;
16       var dados = "";
17       for (i = 0; i < json.catalog.book.length; i++) {
18         var book = json.catalog.book[i]
19         dados = dados + `Id: ${book.id}\n`;
20         dados = dados + `Autor: ${book.author}\n`;
21         dados = dados + `Título: ${book.title}\n`;
22         dados = dados + `Genero: ${book.genre}\n`;
23         dados = dados + `Preço: ${book.price}\n`;
24         dados = dados + `Data de Publicação: ${book.publish_date}\n`;
25         dados = dados + `Descrição: ${book.description}\n\n`;
26       }
27
28       res.setHeader("Content-Type", "text/plain; charset=utf-8");
29       res.end(dados);
30     });
31   } else {
32     res.setHeader("Content-Type", "text/plain; charset=utf-8");
33     res.end("O URL não contém o nome do ficheiro XML para a realização do parse!\n\n Pedido Inválido!");
34   }
35 }
36 }
37 }
```

■ 8.3 JavaScript – REST web service

Integração de Sistemas

- Second endpoint, that uses XPath expression to get all the books, where price is higher than 30.50 and the book's genre is Computer.

- It also builds a string containing the data in html format and returns it to the client.

```

38   ''
39   app.get('/booksXPath', function (req, res) {
40     const queryObject = url.parse(req.url, true).query;
41     if (queryObject["file"] == "books") {
42       fs.readFile("files/books.xml", 'utf8', function (err, data) {
43         var doc = new dom().parseFromString(data)
44         var nodes = xpath.select("/catalog/book[price>30.50 and genre='Computer']", doc)
45         var ids = xpath.select("/catalog/book/@id", doc)
46         var i = 0;
47         var dados = "<h2>RESULTADOS FILTRAGEM XPATH POR:</h2><br><p>price > 30.50 e genre = Computer</p><br>";
48
49         for (i = 0; i < nodes.length; i++) {
50           var bookId = ids[i].value
51           var book = nodes[i];
52           var autor = book.getElementsByTagName('author')[0].childNodes[0].nodeValue;
53           var titulo = book.getElementsByTagName('title')[0].childNodes[0].nodeValue;
54           var genero = book.getElementsByTagName('genre')[0].childNodes[0].nodeValue;
55           var preco = book.getElementsByTagName('price')[0].childNodes[0].nodeValue;
56           var dataPublicacao = book.getElementsByTagName('publish_date')[0].childNodes[0].nodeValue;
57           var descricao = book.getElementsByTagName('description')[0].childNodes[0].nodeValue;
58           dados = dados + `<p>Id: ${bookId}</p>`;
59           dados = dados + `<p>Autor: ${autor}</p>`;
60           dados = dados + `<p>Título: ${titulo}</p>`;
61           dados = dados + `<p>Genero: ${genero}</p>`;
62           dados = dados + `<p>Preço: ${preco}</p>`;
63           dados = dados + `<p>Data de Publicação: ${dataPublicacao}</p>`;
64           dados = dados + `<p>Descrição: ${descricao}</p><br>`;
65         }
66         res.setHeader("Content-Type", "text/html; charset=utf-8");
67         res.end(dados);
68       });
69     }
70     else {
71       res.setHeader("Content-Type", "text/plain; charset=utf-8");
72       res.end("O URL não contém o nome do ficheiro XML para a realização do parse!\n\n Pedido Inválido!");
73     }
  
```

■ 8.3 JavaScript – REST web service

Integração de Sistemas

- The first image, presents the code to create and start the server, configured to listen in the port 3000 in the localhost. The second one is the execution of the created file, through the command line, and the web service running.

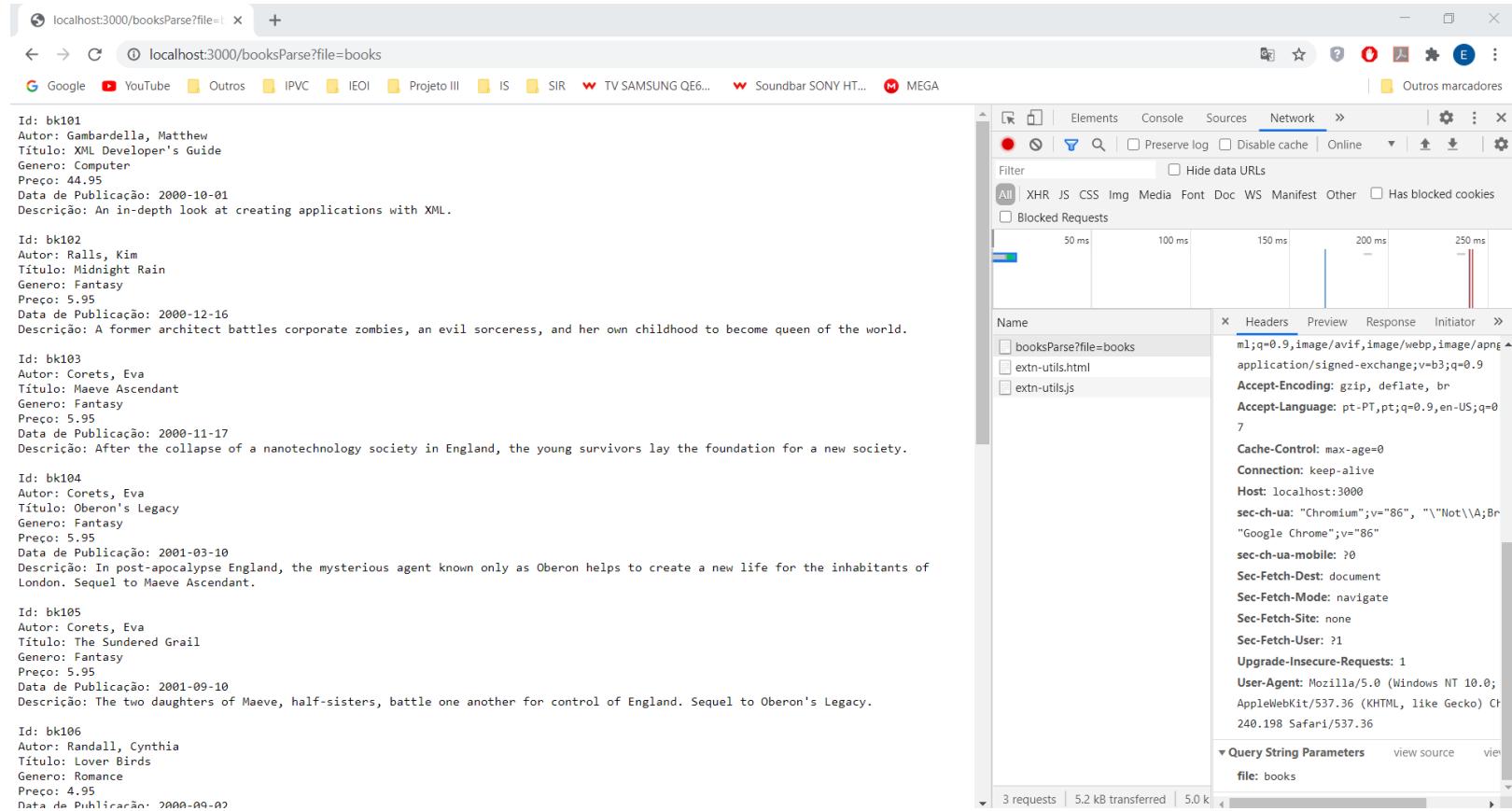
```
77  var server = app.listen(3000, function () {  
78    var host = server.address().address  
79    var port = server.address().port  
80    console.log("Books REST Web Service listening at http://localhost:3000")  
81  })
```

```
Linha de comandos - node index.js  
  
C:\Integração de Sistemas\TP1\ex5_codigo\JavaScript\REST>node index.js  
Books REST Web Service listening at http://localhost:3000
```

■ 8.3 JavaScript – REST web service

Integração de Sistemas

■ Result obtained calling the first endpoint to parse the books.xml file.



localhost:3000/booksParse?file=books

```

Id: bkl01
Autor: Gambardella, Matthew
Título: XML Developer's Guide
Genero: Computer
Preço: 44.95
Data de Publicação: 2000-10-01
Descrição: An in-depth look at creating applications with XML.

Id: bkl02
Autor: Ralls, Kim
Título: Midnight Rain
Genero: Fantasy
Preço: 5.95
Data de Publicação: 2000-12-16
Descrição: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Id: bkl03
Autor: Corets, Eva
Título: Maeve Ascendant
Genero: Fantasy
Preço: 5.95
Data de Publicação: 2000-11-17
Descrição: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.

Id: bkl04
Autor: Corets, Eva
Título: Oberon's Legacy
Genero: Fantasy
Preço: 5.95
Data de Publicação: 2001-03-10
Descrição: In post-apocalypse England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve Ascendant.

Id: bkl05
Autor: Corets, Eva
Título: The Sundered Grail
Genero: Fantasy
Preço: 5.95
Data de Publicação: 2001-09-10
Descrição: The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.

Id: bkl06
Autor: Randall, Cynthia
Título: Lover Birds
Genero: Romance
Preço: 4.95
Data de Publicação: 2000-09-02

```

Network tab details:

- Request URL: localhost:3000/booksParse?file=books
- Request Headers:
 - Content-Type: application/x-www-form-urlencoded
 - Accept: */*
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.102 Safari/537.36
- Response Headers:
 - Content-Type: application/xml
 - Content-Length: 1024
 - Date: Mon, 19 Apr 2021 14:45:21 GMT
 - Server: Apache/2.4.41 (Ubuntu)
 - X-Powered-By: PHP/8.0.12
- Request Body: file=books
- Response Body: XML content (same as above)

■ 8.3 JavaScript – REST web service

Integração de Sistemas

- Result obtained calling the second endpoint to parse and filter the books.xml file with XPath.

localhost:3000/booksXpath?file=books

RESULTADOS FILTRAGEM XPATH POR:

price > 30.50 e genre = Computer

Id: bk101

Autor: Gambardella, Matthew

Titulo: XML Developer's Guide

Genero: Computer

Preço: 44.95

Data de Publicação: 2000-10-01

Descrição: An in-depth look at creating applications with XML.

Id: bk102

Autor: Gambardella, Matthew

Titulo: XML Developer's Guide

Genero: Computer

Preço: 44.95

Data de Publicação: 2000-10-01

Descrição: An in-depth look at creating applications with XML.

Network tab (Chrome DevTools)

Request URL: http://localhost:3000/booksXpath?file=books

Request Method: GET

Status Code: 200 OK

Remote Address: [::]:3000

Referrer Policy: strict-origin-when-cross-origin

Response Headers

- Connection: keep-alive
- Content-Length: 1063
- Content-Type: text/html; charset=utf-8
- Date: Mon, 23 Nov 2020 15:38:33 GMT
- Keep-Alive: timeout=5
- X-Powered-By: Express

Request Headers

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
- Accept-Encoding: gzip, deflate, br
- Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
- Connection: keep-alive
- Host: localhost:3000
- sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome"

3 requests | 3.3 kB transferred | 3.1

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

- In order to create a SOAP web service, it was necessary to create a wsdl file which specifies the web service, defining the operations that it includes, the arguments and the type of call, which in this case is soap, and the names and port where the service can be called.

- The code for the web service was created using the express server and soap packages for Node Js, and a SOAP client to call the created web service, as demonstrated in the next slides.

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

- First part of the Books_ws_wsdl file, which defines the requests and the elements they include.

```

Books_ws_wsdl.wsdl
SOAP > Books_ws_wsdl.wsdl
1  <?xml version="1.0" encoding="UTF-8"?>
2  <wsdl:definitions targetNamespace="http://tempuri.org/"
3      xmlns:s="http://www.w3.org/2001/XMLSchema"
4      xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
5      xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
6      xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
7      xmlns:tns="http://tempuri.org/"
8      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
9      xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
10     xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
11     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
12     <!-- Definição dos pedidos e respostas para o método parse_books --&gt;
13     &lt;wsdl:types&gt;
14         &lt;s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"&gt;
15             &lt;s:element name="BooksParseRequest"&gt;
16                 &lt;s:complexType&gt;
17                     &lt;s:sequence&gt;
18                         &lt;s:element minOccurs="1" maxOccurs="unbounded" name="type" type="s:string"/&gt;
19                     &lt;/s:sequence&gt;
20                 &lt;/s:complexType&gt;
21             &lt;/s:element&gt;
22             &lt;s:element name="BooksParseResponse"&gt;
23                 &lt;s:complexType&gt;
24                     &lt;s:sequence&gt;
25                         &lt;s:element minOccurs="1" maxOccurs="unbounded" name="result" type="s:string"/&gt;
26                     &lt;/s:sequence&gt;
27                 &lt;/s:complexType&gt;
28             &lt;/s:element&gt;
29         &lt;/s:schema&gt;
30     &lt;/wsdl:types&gt;
</pre>

```

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

■ This second part defines the type of communications between the server and client, the operation on the web service , and the way the messages will be transmitted.

```

Books_ws_wsdl.wsdl X
SOAP > Books_ws_wsdl.wsdl
31 | <!-- Definição dos dados e do seu tipo entre o cliente e o servidor -->
32 | <wsdl:message name="BooksParseSoapIn">
33 | | <wsdl:part name="parameters" element="tns:BooksParseRequest"/>
34 | </wsdl:message>
35 | <wsdl:message name="BooksParseSoapOut">
36 | | <wsdl:part name="parameters" element="tns:BooksParseResponse"/>
37 | </wsdl:message>
38 | <wsdl:portType name="BooksParseSoapPort">
39 | | <!-- Definição da operação books_parse, definida no index.js -->
40 | | <wsdl:operation name="BooksParse">
41 | | | <wsdl:input message="tns:BooksParseSoapIn"/>
42 | | | <wsdl:output message="tns:BooksParseSoapOut"/>
43 | | </wsdl:operation>
44 | </wsdl:portType>
45 | <!-- Definição de como as mensagens serão transmitidas -->
46 <wsdl:binding name="BooksParseServiceSoapBinding" type="tns:BooksParseSoapPort">
47 | <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
48 | <wsdl:operation name="BooksParse">
49 | | <soap:operation soapAction="BooksParse" style="document"/>
50 | | <wsdl:input>
51 | | | <soap:body use="literal"/>
52 | | </wsdl:input>
53 | | <wsdl:output>
54 | | | <soap:body use="literal"/>
55 | | </wsdl:output>
56 | </wsdl:operation>
57 </wsdl:binding>
```

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

- The final part creates the definition of the service where the method is going to be binded to.

```
Books_ws_wsdl.wsdl X
SOAP > Books_ws_wsdl.wsdl
58  <wsdl:binding name="BooksParseServiceSoap12Binding" type="tns:BooksParseSoapPort">
59    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
60    <wsdl:operation name="BooksParse">
61      <soap12:operation soapAction="BooksParse" style="document"/>
62      <wsdl:input>
63        <soap12:body use="literal"/>
64      </wsdl:input>
65      <wsdl:output>
66        <soap12:body use="literal"/>
67      </wsdl:output>
68    </wsdl:operation>
69  </wsdl:binding>
70  <!-- Definição do serviço -->
71  <wsdl:service name="BooksParseService">
72    <wsdl:port name="BooksParseServiceSoapPort" binding="tns:BooksParseServiceSoapBinding">
73      <soap:address location="http://localhost:8000/wsdl"/>
74    </wsdl:port>
75    <wsdl:port name="BooksParseServiceSoap12Port" binding="tns:BooksParseServiceSoap12Binding">
76      <soap12:address location="http://localhost:8000/wsdl"/>
77    </wsdl:port>
78  </wsdl:service>
79 </wsdl:definitions>
```

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

■ This first part of the file presents the initial imports necessary for the web service, and the method *books_parse*, which depending on the argument given in the call, will execute the parse or filter and parse the books.xml file.

```
JS index.js ●
SOAP > JS index.js > ...
1 var soap = require('soap');
2 var express = require('express');
3 var fs = require('fs');
4 var parser = require('xml2json-light');
5 var xpath = require('xpath')
6 var dom = require('xmldom').DOMParser
7
8 function books_parse(args) {
9   if(args.type == "parse") {
10     var data = fs.readFileSync( "files/books.xml", 'utf8');
11     var json = parser.xml2json(data);
12     var dados = "<h2>PARSE DO FICHEIRO - books.xml</h2><br>";
13     var i = 0;
14     for(i = 0; i < json.catalog.book.length; i++) {
15       var book = json.catalog.book[i]
16       dados = dados + `<p>Id: ${book.id}</p>`;
17       dados = dados + `<p>Autor: ${book.author}</p>`;
18       dados = dados + `<p>Título: ${book.title}</p>`;
19       dados = dados + `<p>Genero: ${book.genre}</p>`;
20       dados = dados + `<p>Preço: ${book.price}</p>`;
21       dados = dados + `<p>Data de Publicação: ${book.publish_date}</p>`;
22       dados = dados + `<p>Descrição: ${book.description}</p><br>`;
23     }
24     return {
25       result: dados
26     }
27 }
```

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

```

29  if(args.type == "xpath") {
30      var data = fs.readFileSync( "files/books.xml", 'utf8');
31      var doc = new dom().parseFromString(data)
32      var nodes = xpath.select("/catalog/book[genre='Fantasy' and contains(author, 'Eva')]", doc)
33      var ids = xpath.select("/catalog/book/@id", doc)
34      var i = 0;
35      var dados = "<h2>XPATH - books.xml</h2><p>Filtragem por livros do género <i>Fantasy</i>,\\
36      e onde o nome do autor contém a string, <i>Eva</i></p><br>";
37      for(i = 0; i < nodes.length; i++) {
38          var bookId = ids[i].value
39          var book = nodes[0];
40          var autor = book.getElementsByTagName('author')[0].childNodes[0].nodeValue;
41          var titulo = book.getElementsByTagName('title')[0].childNodes[0].nodeValue;
42          var genero = book.getElementsByTagName('genre')[0].childNodes[0].nodeValue;
43          var preco = book.getElementsByTagName('price')[0].childNodes[0].nodeValue;
44          var dataPublicacao = book.getElementsByTagName('publish_date')[0].childNodes[0].nodeValue;
45          var descricao = book.getElementsByTagName('description')[0].childNodes[0].nodeValue;
46          dados = dados + `<p>Id: ${bookId}</p>`\\
47          dados = dados + `<p>Autor: ${autor}</p>`\\
48          dados = dados + `<p>Título: ${titulo}</p>`\\
49          dados = dados + `<p>Genero: ${genero}</p>`\\
50          dados = dados + `<p>Preço: ${preco}</p>`\\
51          dados = dados + `<p>Data de Publicação: ${dataPublicacao}</p>`\\
52          dados = dados + `<p>Descrição: ${descricao}</p><br>`\\
53      }
54      return {
55          result: dados
56      }
57  }
58 }
59 }
```

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

- Service object created and the wsdl file readed, configuring the *book_parse* method to be executed when the server is accessed in the port 8000 on the localhost.

```

JS index.js •
SOAP > JS index.js > ...
59
60  var serviceObject = {
61    BooksParseService: {
62      BooksParseServiceSoapPort: {
63        BooksParse: books_parse
64      },
65      BooksParseServiceSoap12Port: {
66        BooksParse: books_parse
67      }
68    }
69  };
70
71  var xml = fs.readFileSync('Books_ws_wsdl.wsdl', 'utf8');
72  var app = express();
73
74  app.get('/', function (req, res) {
75    res.send('SOAP webservice - Parse de ficheiros XML e XPATH<br />');
76  })
77
78  var port = 8000;
79  app.listen(port, function () {
80    console.log('Listening on port ' + port);
81    var wsdl_path = "/wsdl";
82    soap.listen(app, wsdl_path, serviceObject, xml);
83    console.log("Check http://localhost:" + port + wsdl\_path + "?wsdl to see if the service is working");
84  });

```

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

- Client code, using the soap and express libraries to call the web service and present the results, the first endpoint for the parsing of the books.xml file.

```
JS client.js  x

SOAP > JS client.js > app.get('/callBooksParse') callback > soap.createClient() callback
1  var express = require('express');
2  var app = express();
3  var soap = require('soap');
4  var url = 'http://localhost:8000/wsdl?wsdl';
5
6  app.get('/callBooksParse', function (req, res) {
7    soap.createClient(url, function (err, client) {
8      if (err){
9        throw err;
10     }
11     var args = {
12       type: 'parse'
13     }
14     client.BooksParse(args, function (err, result) {
15       if (err) {
16         throw err;
17       }
18       else {
19         res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});
20         res.end(result["result"][0]);
21       }
22     });
23   });
24 })
25 }
```

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

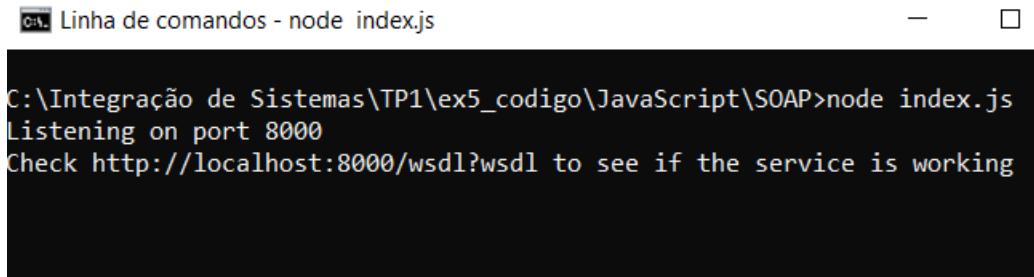
■ Second endpoint where the web service is called specifying the type of action as XPath, presenting the results as html.

```
JS client.js ×
SOAP > JS client.js > app.get('/callBooksParse') callback > soap.createClient() callback
25
26   app.get('/callBooksXPath', function (req, res) {
27     soap.createClient(url, function (err, client) {
28       if (err){
29         throw err;
30       }
31       var args = {
32         type: 'xpath'
33       }
34       client.BooksParse(args, function (err, result) {
35         if (err) {
36           throw err;
37         }
38         else {
39           res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});
40           res.end(result["result"][0]);
41         }
42       });
43     });
44   })
45
46   var server = app.listen(3000, function () {
47     var host = server.address().address
48     var port = server.address().port
49     console.log("Example app listening at http://localhost:3000")
50   })
```

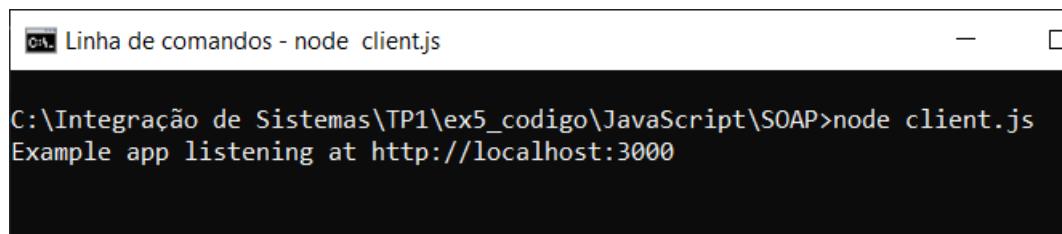
■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

- Running the server (image on top), running the client (image on the bottom).



```
Linha de comandos - node index.js
C:\Integração de Sistemas\TP1\ex5_codigo\JavaScript\SOAP>node index.js
Listening on port 8000
Check http://localhost:8000/wsdl?wsdl to see if the service is working
```



```
Linha de comandos - node client.js
C:\Integração de Sistemas\TP1\ex5_codigo\JavaScript\SOAP>node client.js
Example app listening at http://localhost:3000
```

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

■ Results obtained calling the webservice to parse the books.xml file.

localhost:3000/callBooksParse

PARSE DO FICHEIRO - books.xml

Id: bk101
 Autor: Gambardella, Matthew
 Titulo: XML Developer's Guide
 Genero: Computer
 Preço: 44.95
 Data de Publicação: 2000-10-01
 Descrição: An in-depth look at creating applications with XML.

Id: bk102
 Autor: Ralls, Kim
 Titulo: Midnight Rain
 Genero: Fantasy
 Preço: 5.95
 Data de Publicação: 2000-12-16
 Descrição: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Id: bk103
 Autor: Corets, Eva

Network tab in browser developer tools showing the request and response details for the callBooksParse API call.

Name	Headers	Preview	Response	Initiator	Timing	Cookies
callBooksParse						
extn-utils.html						
extn-utils.js						

Request Headers

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
- Accept-Encoding: gzip, deflate, br
- Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
- Connection: keep-alive
- Host: localhost:3000
- sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome";v="86"

Response Headers

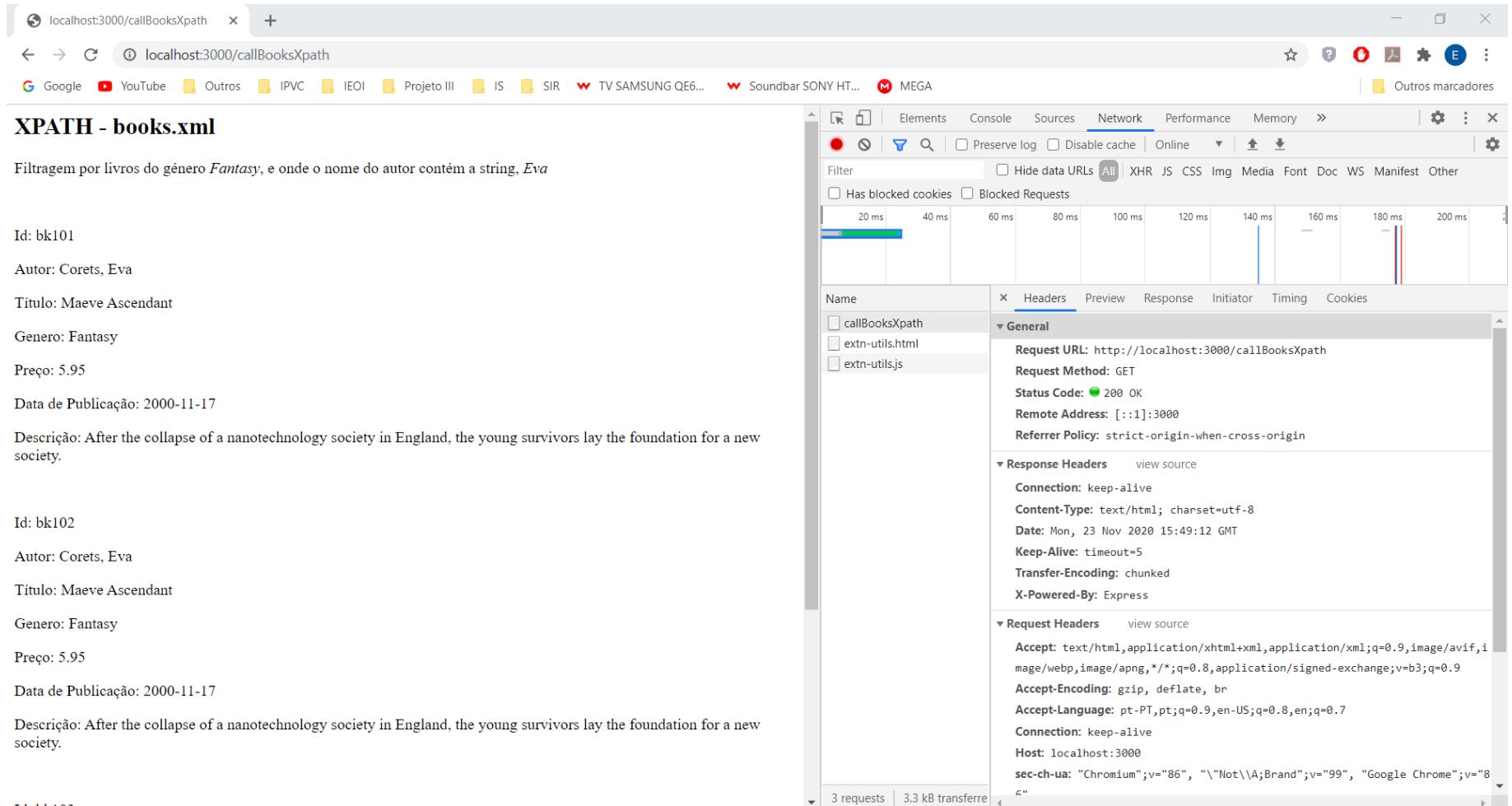
- Connection: keep-alive
- Content-Type: text/html; charset=utf-8
- Date: Mon, 23 Nov 2020 15:48:36 GMT
- Keep-Alive: timeout=5
- Transfer-Encoding: chunked
- X-Powered-By: Express

Request URL: http://localhost:3000/callBooksParse
Request Method: GET
Status Code: 200 OK
Remote Address: [::1]:3000
Referrer Policy: strict-origin-when-cross-origin

■ 8.4 JavaScript – SOAP web service

Integração de Sistemas

■ Results obtained calling the webservice to parse and filter the books.xml file with Xpath.



XPATH - books.xml

Filtragem por livros do género *Fantasy*, e onde o nome do autor contém a string, *Eva*

Id: bk101

Autor: Corets, Eva

Título: Maeve Ascendant

Genero: Fantasy

Preço: 5.95

Data de Publicação: 2000-11-17

Descrição: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.

Id: bk102

Autor: Corets, Eva

Título: Maeve Ascendant

Genero: Fantasy

Preço: 5.95

Data de Publicação: 2000-11-17

Descrição: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.

The screenshot shows a browser window with the URL `localhost:3000/callBooksXpath`. The page content displays two book entries, bk101 and bk102, filtered by genre *Fantasy* and author name containing *Eva*. To the right of the browser is the Chrome developer tools Network tab, which shows a single GET request to the same URL. The request details panel shows the following information:

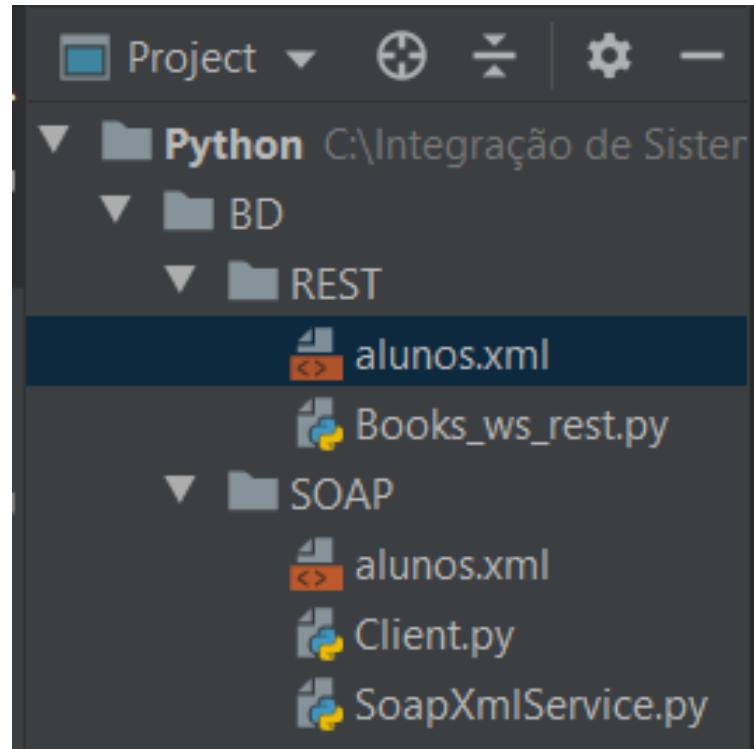
- Request URL: `http://localhost:3000/callBooksXpath`
- Request Method: GET
- Status Code: 200 OK
- Remote Address: `[::1]:3000`
- Referrer Policy: strict-origin-when-cross-origin
- Response Headers:
 - Connection: keep-alive
 - Content-Type: text/html; charset=utf-8
 - Date: Mon, 23 Nov 2020 15:49:12 GMT
 - Keep-Alive: timeout=5
 - Transfer-Encoding: chunked
 - X-Powered-By: Express
- Request Headers:
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
 - Accept-Encoding: gzip, deflate, br
 - Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
 - Connection: keep-alive
 - Host: localhost:3000
 - sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome";v="86"

■ 9 Python SOAP and REST web services to read data from MySQL database**Integração de Sistemas**

- For the REST and SOAP web services involving the access to a MySQL database, the same libraries were used, but additionally it was imported the MySQL connector for Python, allowing the creation of a connection to the database running on the localhost in the port 3306.
- The following slide demonstrates the project's structure, created with the SOAP and REST web services.

■ 9 Python SOAP and REST web services to read data from MySQL database

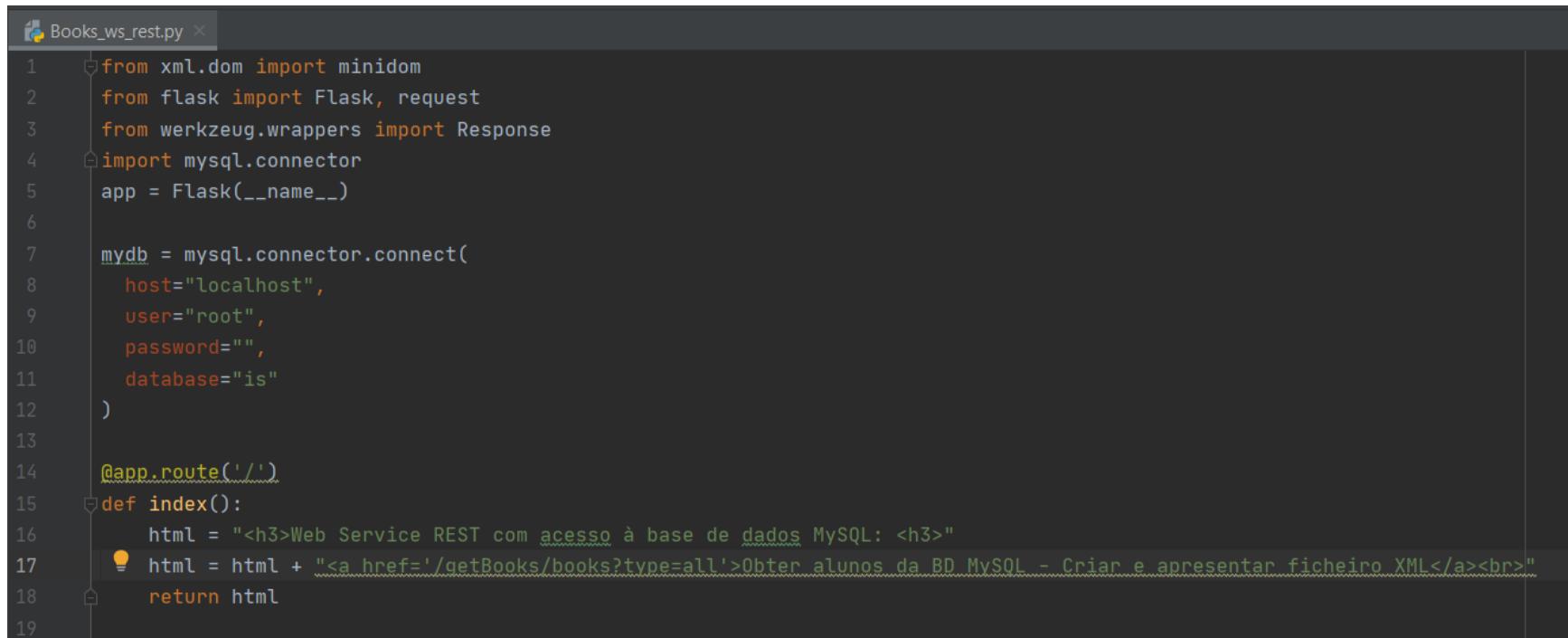
Integração de Sistemas



■ 9.1 Python – REST web service

Integração de Sistemas

- Necessary imports used in the web service, definition of the connection to the MySQL database and the first endpoint of the web service.



```
Books_ws_rest.py ×
1  from xml.dom import minidom
2  from flask import Flask, request
3  from werkzeug.wrappers import Response
4  import mysql.connector
5  app = Flask(__name__)
6
7  mydb = mysql.connector.connect(
8      host="localhost",
9      user="root",
10     password="",
11     database="is"
12 )
13
14 @app.route('/')
15 def index():
16     html = "<h3>Web Service REST com acesso à base de dados MySQL: </h3>"
17     html = html + "<a href='/getBooks/books?type=all'>Obter alunos da BD MySQL - Criar e apresentar ficheiro XML</a><br>"
18     return html
19
```

■ 9.1 Python – REST web service

Integração de Sistemas

- Second endpoint with the execution of an SQL query to obtain all the students in the table *alunos*, as well as the creation of the tags to construct the XML file, which is created in the same folder where the *books_ws_rest.py* is defined.

- At the end, the data is returned to the client and printed as XML.

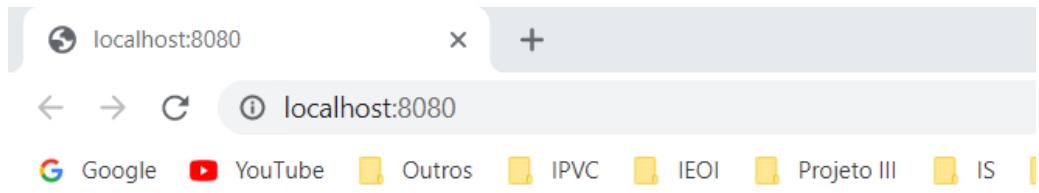
```
Books_ws_rest.py ×
20     @app.route('/getAlunos/alunos', methods=['GET'])
21     def getAlunos():
22         selectType = request.args.get('type')
23         if selectType.upper() == "ALL":
24             mycursor = mydb.cursor()
25             mycursor.execute("SELECT * FROM alunos")
26             myresult = mycursor.fetchall()
27             doc = minidom.Document()
28             tag_alunos = doc.createElement('alunos')
29             for aluno in myresult:
30                 id_aluno = str(aluno[0])
31                 nome = aluno[1]
32                 tag_aluno = doc.createElement('aluno')
33
34                 tag_id = doc.createElement('id')
35                 content_id = doc.createTextNode(id_aluno)
36                 tag_id.appendChild(content_id)
37
38                 tag_nome = doc.createElement('nome')
39                 content_nome = doc.createTextNode(nome)
40                 tag_nome.appendChild(content_nome)
41
42                 tag_aluno.appendChild(tag_id)
43                 tag_aluno.appendChild(tag_nome)
44
45             tag_alunos.appendChild(tag_aluno)
46
47             doc.appendChild(tag_alunos)
48             string_xml = doc.toprettyxml(indent="\t")
49             caminho_ficheiro = "alunos.xml"
50             with open(caminho_ficheiro, "w") as f:
51                 f.write(string_xml)
```

■ 9.1 Python – REST web service

Integração de Sistemas

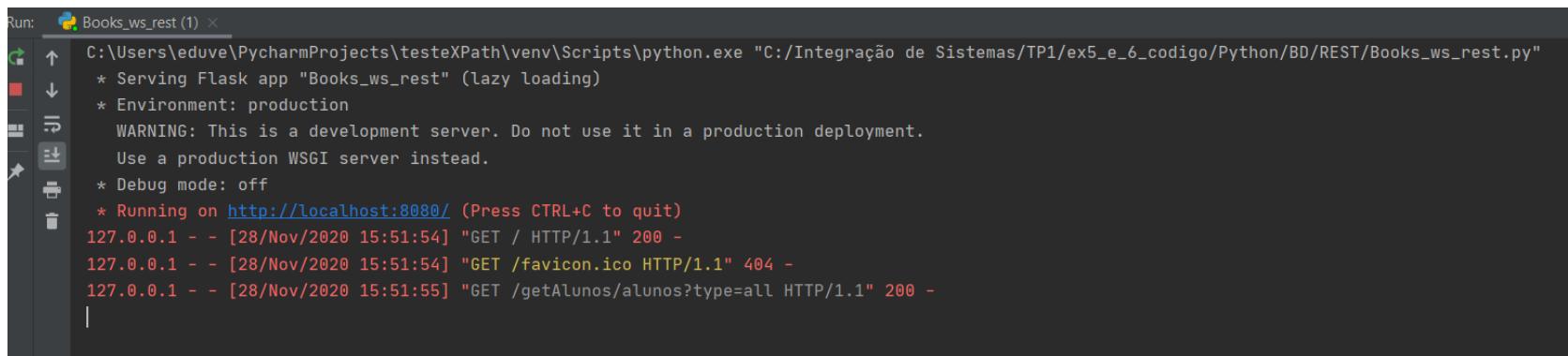
The image on top demonstrates the response returned to the client, the second, the link to call the web service, and the final image the call registered in the web service when it is called.

```
52         return Response(string_xml, mimetype='text/xml')
53
54 ► if __name__ == "__main__":
55     app.run(host='localhost', port=8080)
56
```



Web Service REST com acesso à base de dados MySQL:

Obter alunos da BD MySQL - Criar e apresentar ficheiro XML

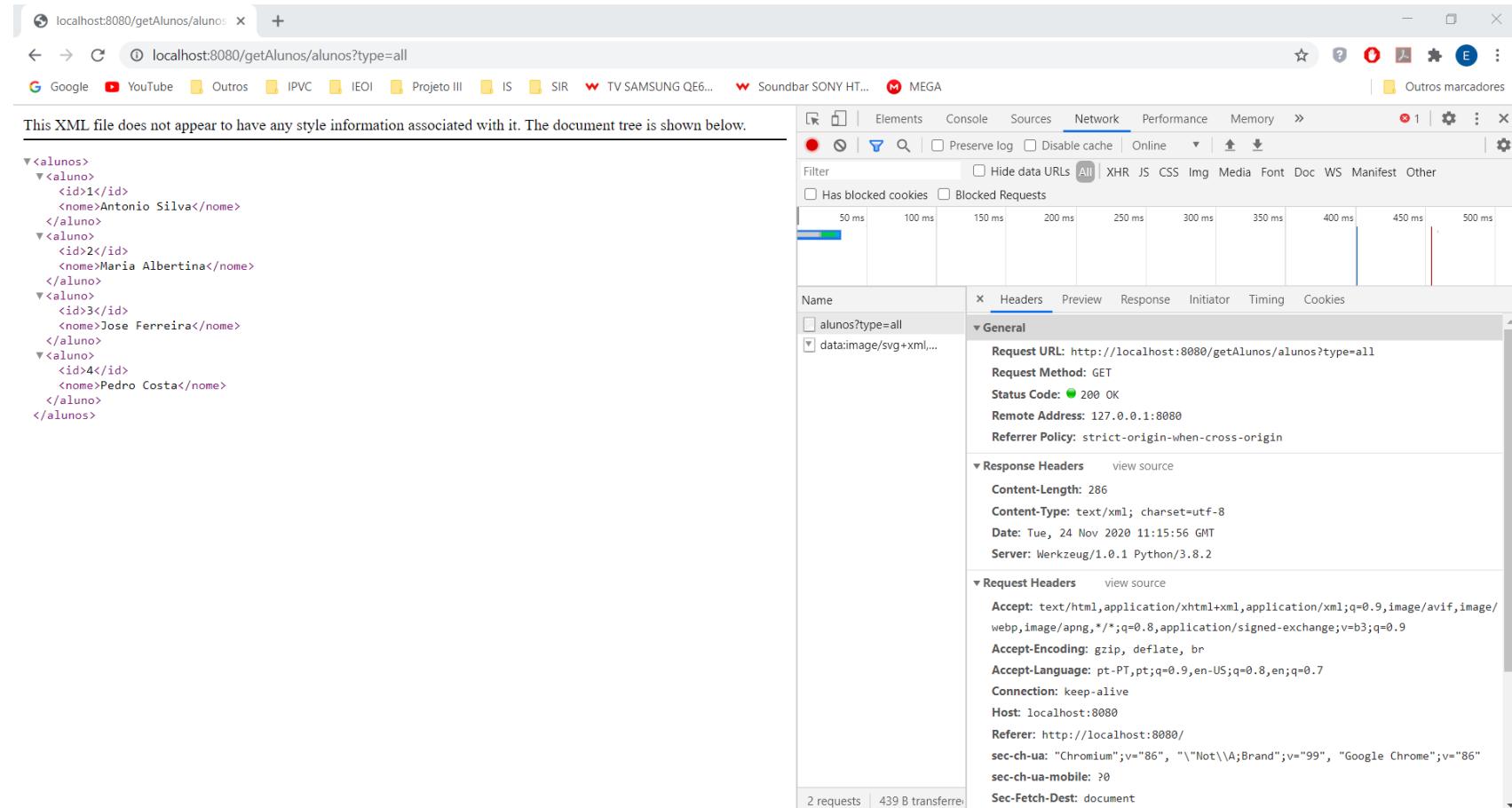

 A screenshot of the PyCharm IDE showing the run output for the file "Books_ws_rest.py". The output shows the server starting and listening on "http://localhost:8080/". It also shows three log entries from the terminal:


```
* Serving Flask app "Books_ws_rest" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://localhost:8080/ (Press CTRL+C to quit)
127.0.0.1 - - [28/Nov/2020 15:51:54] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Nov/2020 15:51:54] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [28/Nov/2020 15:51:55] "GET /getAlunos/alunos?type=all HTTP/1.1" 200 -
```

■ 9.1 Python – REST web service

Integração de Sistemas

■ Result obtained by calling the web service



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<alunos>
  <aluno>
    <id>1</id>
    <nome>Antonio Silva</nome>
  </aluno>
  <aluno>
    <id>2</id>
    <nome>Maria Albertina</nome>
  </aluno>
  <aluno>
    <id>3</id>
    <nome>Jose Ferreira</nome>
  </aluno>
  <aluno>
    <id>4</id>
    <nome>Pedro Costa</nome>
  </aluno>
</alunos>

```

The screenshot shows a browser window with the URL `localhost:8080/getAlunos/alunos`. The developer tools Network tab is open, showing a GET request to the same URL. The request details pane shows the following information:

- Request URL:** `http://localhost:8080/getAlunos/alunos?type=all`
- Request Method:** GET
- Status Code:** 200 OK
- Remote Address:** 127.0.0.1:8080
- Referrer Policy:** strict-origin-when-cross-origin
- Content-Length:** 286
- Content-Type:** text/xml; charset=utf-8
- Date:** Tue, 24 Nov 2020 11:15:56 GMT
- Server:** Werkzeug/1.0.1 Python/3.8.2

The Headers, Response, Initiator, Timing, and Cookies tabs are also visible in the developer tools.

■ 9.2 Python – SOAP web service

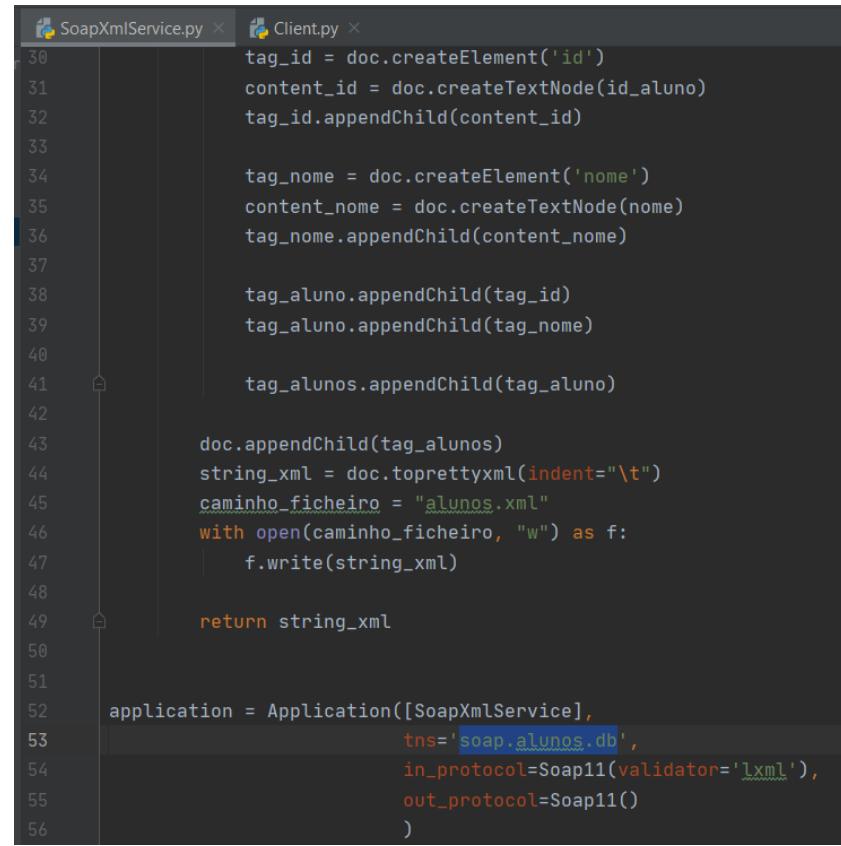
Integração de Sistemas

- For the SOAP web service the code is similar to the REST web service, being the use of `spyne`, creating a server and registering the web service with a *tns*, the only difference in relation to the REST web service, as demonstrated by the following images.

```

57 ► if __name__ == '__main__':
58     from wsgiref.simple_server import make_server
59
60     wsgi_app = WsgiApplication(application)
61     server = make_server('localhost', 8000, wsgi_app)
62     server.serve_forever()
63

```



```

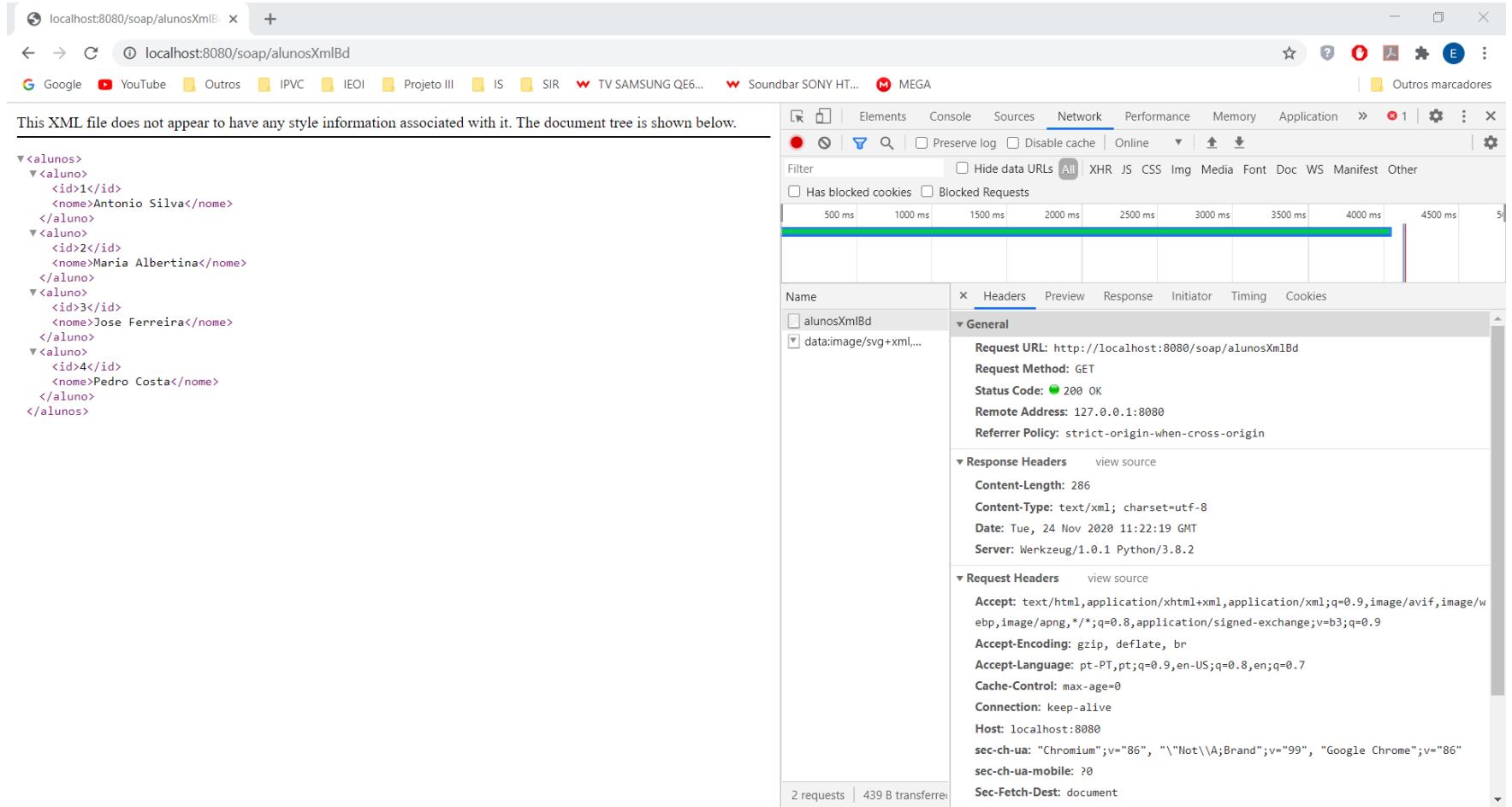
SoapXmlService.py x Client.py x
30 tag_id = doc.createElement('id')
31 content_id = doc.createTextNode(id_aluno)
32 tag_id.appendChild(content_id)
33
34 tag_nome = doc.createElement('nome')
35 content_nome = doc.createTextNode(nome)
36 tag_nome.appendChild(content_nome)
37
38 tag_aluno.appendChild(tag_id)
39 tag_aluno.appendChild(tag_nome)
40
41 tag_alunos.appendChild(tag_aluno)
42
43 doc.appendChild(tag_alunos)
44 string_xml = doc.toprettyxml(indent="\t")
45 caminho_ficheiro = "alunos.xml"
46 with open(caminho_ficheiro, "w") as f:
47     f.write(string_xml)
48
49
50
51
52 application = Application([SoapXmlService],
53                             tns='soap.alunos.db',
54                             in_protocol=Soap11(validation='XML'),
55                             out_protocol=Soap11()
56 )

```

■ 9.2 Python – SOAP web service

Integração de Sistemas

- Result obtained by calling the web service with the client.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<alunos>
  <aluno>
    <id>1</id>
    <nome>Antonio Silva</nome>
  </aluno>
  <aluno>
    <id>2</id>
    <nome>Maria Albertina</nome>
  </aluno>
  <aluno>
    <id>3</id>
    <nome>Jose Ferreira</nome>
  </aluno>
  <aluno>
    <id>4</id>
    <nome>Pedro Costa</nome>
  </aluno>
</alunos>

```

The Network tab of the developer tools shows the following request details:

- Request URL:** http://localhost:8080/soap/alunosXmlBd
- Request Method:** GET
- Status Code:** 200 OK
- Remote Address:** 127.0.0.1:8080
- Referrer Policy:** strict-origin-when-cross-origin
- Content-Length:** 286
- Content-Type:** text/xml; charset=utf-8
- Date:** Tue, 24 Nov 2020 11:22:19 GMT
- Server:** Werkzeug/1.0.1 Python/3.8.2

The Request Headers are:

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
- Accept-Encoding: gzip, deflate, br
- Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7
- Cache-Control: max-age=0
- Connection: keep-alive
- Host: localhost:8080
- sec-ch-ua: "Chromium";v="86", "\Not\A;Brand";v="99", "Google Chrome";v="86"
- sec-ch-ua-mobile: ?0
- Sec-Fetch-Dest: document

■ 9.2 Python – SOAP web service

Integração de Sistemas

- The image on top demonstrates the requests done on the client code, and the image on the bottom the requests and responses executed by the web service.

```
* Running on http://localhost:8080/ (Press CTRL+C to quit)
127.0.0.1 - - [24/Nov/2020 11:21:41] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Nov/2020 11:22:11] "GET /soap/alunosXmlBd HTTP/1.1" 200 -
127.0.0.1 - - [24/Nov/2020 11:22:19] "GET /soap/alunosXmlBd HTTP/1.1" 200 -
```

```
C:\Users\eduve\PycharmProjects\testeXPath\venv\Scripts\python.exe "C:/Integração de Sistemas/TP1/ex5_e_6_codigo/Python/BD/SOAP/SoapXmlService.py"
127.0.0.1 - - [24/Nov/2020 11:22:09] "GET /soap.alunos.db?wsdl HTTP/1.1" 200 2402
127.0.0.1 - - [24/Nov/2020 11:22:11] "POST /soap.alunos.db HTTP/1.1" 200 738
127.0.0.1 - - [24/Nov/2020 11:22:17] "GET /soap.alunos.db?wsdl HTTP/1.1" 200 2402
127.0.0.1 - - [24/Nov/2020 11:22:19] "POST /soap.alunos.db HTTP/1.1" 200 738
```

■ 10. Conclusion

■ To conclude it is possible to state that this practical work has helped to increase our knowledge about the implementation of webservices in JAVA, PHP, JavaScript and Python, as well as utilizing different XML parsers to obtain and read the content of several XML files, with different sizes, and the implications of doing so in large files. This knowledge allowed us to solidify theoretical concepts learned during the classes and apply them practically, such as XPath and the use of its syntax. Additionally it allowed us to use connection methods to a MySQL database running on a local server and obtain data from it and process it in a webservice call.

■ 11. Bibliography and Web References

[1]. TutorialsPoint . 2020. Java XML - Parsers. Available at:

https://www.tutorialspoint.com/java_xml/java_xml_parsers.htm;

[2]. PhpMyAdmin Contributors. 2020. Documentation. Available at:

<https://www.phpmyadmin.net/docs/>;

[3]. W3Schools . 2020. XML Validator. Available at:

https://www.w3schools.com/xml/xml_validator.asp;

[4]. Arskom Ltd. 2020. What is Spyne?. Available at:

<http://spyne.io/#inprot=Soap11&outprot=XmlDocument&s=rpc&tpt=WsgiApplication&validator=true>;

[5]. LinuxConfig.org . 2020. Deployment of an Example Application into Apache Tomcat Container. Available at:

<https://linuxconfig.org/deployment-of-an-example-application-into-apache-tomcat-container>;

[6]. Python Software Foundation. 2020. Flask-Spyne 0.3.1. Available at:

<https://pypi.org/project/Flask-Spyne/>;

■ 11. Bibliography and Web References

[7]. Onejohi. 2020. Building a simple REST API with NodeJS and Express. Available at:

<https://medium.com/@onejohi/building-a-simple-rest-api-with-nodejs-and-express-da6273ed7ca9;>

[8]. Filestack. 2020. Common problems with large file uploads. Available at:

<https://blog.filestack.com/thoughts-and-knowledge/common-problems-with-large-file-uploads/>;

[9]. Filestack. 2019. How To Upload Large Files. Available at:

<https://blog.filestack.com/thoughts-and-knowledge/how-to-upload-large-files/>;

[10]. James Kleeh. 2019. Code sample: Async parsing. Available at:

<https://github.com/FasterXML/aalto-xml/wiki/Code-sample:-Async-parsing;>

[11]. The PHP Group. 2020. XMLReader. Available at:

<https://www.php.net/manual/en/book.xmlreader.php>;



Instituto Politécnico
de Viana do Castelo



Instituto Politécnico
de Viana do Castelo

GRADUATION IN INFORMATICS ENGINEERING



Escola Superior
de Tecnologia e Gestão

Practical Work nº1

**Parsing of XML Files using REST and
SOAP via JAVA, PHP and PYTHON**

■ Course Unit: Integração de Sistemas

Eduardo Eiras Nº 21484

■ Coordinator: Prof. Doctor Jorge Ribeiro

Instituto Politécnico de Viana do Castelo
Escola Superior de Tecnologia e Gestão
www.ipvc.pt