

Licenciatura em Engenharia Informática

Processamento Estruturado de informação

BikeOnTrack

Docentes: Bruno Oliveira

Marco Gomes

Alunos: João Marques 8170200

Pedro Pinto 8170262

Ricardo Ferreira 8170279

Conteúdo

Introdução	3
XML Schema	4
http://www.BikeOnTrack.pt/Sales	5
http://www.BikeOnTrack.pt/Loja	7
http://www.BikeOnTrack.pt/Cliente	8
http://www.BikeOnTrack.pt/SaleLine	9
http://www.BikeOnTrack.pt/Produto	10
http://www.BikeOnTrack.pt/CurrencyDetails	11
http://www.BikeOnTrack.pt/ReceiptInfo	12
http://www.BikeOnTrack.pt/CommonTypes	13
Exemplo XML	15
Conclusão	18
Referências	19

Introdução

O presente trabalho surge no âmbito da unidade curricular de Processamento Estruturado de Informação, referente á Licenciatura em Engenharia Informática da Escola Superior de Tecnologia e Gestão.

Este trabalho tem como principal objetivo o desenvolvimento de um vocabulário XML capaz de captar todos os requisitos relacionados com o armazenamento da informação relacionada com a auditoria das vendas realizadas em cada loja num determinado período, para minimizar o problema da BikeOnTrack.

Desta forma, para resolver o problema, utilizando os objetivos identificados para a elaboração do vocabulário decidimos optar pela seguinte abordagem:

Decidimos optar por dividir o trabalho em vários pontos principais de forma a facilitar o entendimento e modificação dos documentos, como por exemplo:

- Exercício
 - Vendas
 - Loja
 - Cliente
 - Linhas de Venda
 - Produto
 - Moeda
 - Data do exercício

XML Schema

NameSpace	http://www.BikeOnTrack.pt/Exercise
Propriedades	elementFormDefault="qualified"
Nome do Schema	Exercise.xsd

NameSpace	http://www.BikeOnTrack.pt/Exercise
Descrição	<p>Temos presente um ComplexType com 2 elementos: salesTotal, information. O elemento salesTotal e ele mesmo um ComplexType, contendo dentro de si uma referencia ao schema http://www.BikeOnTrack.pt/Sales, optamos por esta abordagem de forma a que a alteração futura destes dados seja mais facilmente aplicada, podendo assim ser criado um xml para cada venda. Este elemento está ainda condicionado a existir no mínimo uma vez.</p> <p>O elemento information e ele mesmo um ComplexType, contendo dentro de si um elemento infoType, que e do tipo ri:info, este tipo foi importado através do schema http://www.BikeOnTrack.pt/ReceiptInfo sendo este condicionado a existir uma única vez.</p>
Elementos filho	e:salesTotal, e:information
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/Exercise" targetNamespace="http://www.BikeOnTrack.pt/Exercise" xmlns:s="http://www.BikeOnTrack.pt/Sales" xmlns:ri="http://www.BikeOnTrack.pt/ReceiptInfo" elementFormDefault="qualified"> <xs:import schemaLocation="Sales.xsd" namespace="http://www.BikeOnTrack.pt/Sales"/> <xs:import schemaLocation="ReceiptInfo.xsd" namespace="http://www.BikeOnTrack.pt/ReceiptInfo"/> <xs:element name="exerciseType"> <xs:complexType> <xs:sequence> <xs:element name="salesTotal"> <xs:complexType> <xs:sequence minOccurs="1" maxOccurs="unbounded"> <xs:element ref="s:saleType"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="information"> <xs:complexType> <xs:sequence> <xs:element name="infoType" type="ri:info" minOccurs="1" maxOccurs="1"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:schema> </pre>

NameSpace	http://www.BikeOnTrack.pt/Sales
Propriedades	elementFormDefault="qualified"
Nome do Schema	Sales.xsd

NameSpace	http://www.BikeOnTrack.pt/Sales
Descrição	<p>Temos presente um ComplexType com 9 elementos: id, lojaInfo, clientInfo, orderDate, line, currency, taxAmt, subtotal, total.</p> <p>O elemento lojaInfo e ele mesmo um ComplexType, contendo dentro de si um elemento lojaType, que e do tipo l:loja, este tipo foi importado através do schema http://www.BikeOnTrack.pt/Loja sendo este condicionado a existir uma única vez.</p> <p>O elemento clientInfo é ele mesmo um ComplexType, contendo dentro de si um elemento clientType, que é do tipo c:cliente, este tipo foi importado através do schema http://www.BikeOnTrack.pt/Cliente sendo este condicionado a existir uma única vez.</p> <p>O elemento line é ele mesmo um ComplexType, contendo dentro de si uma referencia ao schema http://www.BikeOnTrack.pt/SaleLine, optamos por esta abordagem de forma a que a alteração futura destes dados seja mais facilmente aplicada, podendo assim ser criado um xml para cada linha de venda. Este elemento esta ainda condicionado a existir no mínimo uma vez.</p> <p>O elemento currency é ele mesmo um ComplexType, contendo dentro de si um elemento currencyType, que e do tipo cd:currencydetails, este tipo foi importado através do schema http://www.BikeOnTrack.pt/CurrencyDetails sendo este condicionado a existir no máximo uma vez.</p>
Elementos filho	l:id, l:name, l:nif, l:morada e l:contactos
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/Sales" targetNamespace="http://www.BikeOnTrack.pt/Sales" xmlns:ct="http://www.BikeOnTrack.pt/CommonTypes" xmlns:c="http://www.BikeOnTrack.pt/Cliente" xmlns:l="http://www.BikeOnTrack.pt/Loja" xmlns:sl="http://www.BikeOnTrack.pt/SaleLine" xmlns:cd="http://www.BikeOnTrack.pt/CurrencyDetails" elementFormDefault="qualified"> <xs:import schemaLocation="CommonTypes.xsd" namespace="http://www.BikeOnTrack.pt/CommonTypes"/> <xs:import schemaLocation="SaleLine.xsd" namespace="http://www.BikeOnTrack.pt/SaleLine"/> <xs:import schemaLocation="Cliente.xsd" namespace="http://www.BikeOnTrack.pt/Cliente"/> <xs:import schemaLocation="Loja.xsd" namespace="http://www.BikeOnTrack.pt/Loja"/> <xs:import schemaLocation="CurrencyDetails.xsd" namespace="http://www.BikeOnTrack.pt/CurrencyDetails"/> </pre>

Estrutura XSD	<pre> <xs:element name="saleType"> <xs:complexType> <xs:sequence> <xs:element name="id" type="ct:id"/> <xs:element name="lojaInfo"> <xs:complexType> <xs:sequence> <xs:element name="lojaType" type="l:loja" minOccurs="1" maxOccurs="1"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="clientInfo"> <xs:complexType> <xs:sequence> <xs:element name="clientType" type="c:cliente" minOccurs="1" maxOccurs="1"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="orderDate" type="xs:dateTime"/> <xs:element name="line"> <xs:complexType> <xs:sequence minOccurs="1" maxOccurs="unbounded"> <xs:element ref="sl:saleline"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="currency"> <xs:complexType> <xs:sequence> <xs:element name="currencyType" type="cd:currencydetails" minOccurs="0" maxOccurs="1"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="taxAmt" type="ct:valor"/> <xs:element name="subTotal" type="ct:valor"/> <xs:element name="total" type="ct:valor"/> </xs:sequence> </xs:complexType> </xs:element> </xs:schema> </pre>
Dependências em relação a outros elementos (identificar schema)	Exercise

NameSpace	http://www.BikeOnTrack.pt/Loja
Propriedades	elementFormDefault="qualified"
Nome do Schema	Loja.xsd

NameSpace	http://www.BikeOnTrack.pt/Loja
Descrição	Temos presente um ComplexType com 5 elementos: id, name, nif, morada e contacto. Os quais o type e especificado no xsd CommonTypes.
Elementos filho	l:id, l:name, l:nif, l:morada e l:contacto
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/Loja" targetNamespace="http://www.BikeOnTrack.pt/Loja" xmlns:ct="http://www.BikeOnTrack.pt/CommonTypes" elementFormDefault="qualified"> <xs:import schemaLocation="CommonTypes.xsd" namespace="http://www.BikeOnTrack.pt/CommonTypes"/> <xs:complexType name="loja"> <xs:sequence> <xs:element name="id" type="ct:id"/> <xs:element name="name" type="ct:texto"/> <xs:element name="nif" type="ct:nif"/> <xs:element name="morada" type="ct:morada"/> <xs:element name="contacto" type="ct:telefone"/> </xs:sequence> </xs:complexType> </xs:schema> </pre>
Dependências em relação a outros elementos (identificar schema)	s:Sales

Namespace	http://www.BikeOnTrack.pt/Cliente
Propriedades	elementFormDefault="qualified"
Nome do Schema	Cliente.xsd

Namespace	http://www.BikeOnTrack.pt/Cliente
Descrição	Temos presente um ComplexType com 3 elementos: id, nome e nif. Os quais o type e especificado no xsd CommonTypes.
Elementos filho	c:id, c:nome, c:nif.
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/Cliente" targetNamespace="http://www.BikeOnTrack.pt/Cliente" xmlns:ct="http://www.BikeOnTrack.pt/CommonTypes" elementFormDefault="qualified"> <xs:import schemaLocation="CommonTypes.xsd" namespace="http://www.BikeOnTrack.pt/ CommonTypes"/> <xs:complexType name="cliente"> <xs:sequence> <xs:element name="id" type="ct:id"/> <xs:element name="nome" type="ct:texto"/> <xs:element name="nif" type="ct:nif"/> </xs:sequence> </xs:complexType> </xs:schema> </pre>
Dependências em relação a outros elementos (identificar schema)	s:Sales

NameSpace	http://www.BikeOnTrack.pt/SaleLine
Propriedades	elementFormDefault="qualified"
Nome do Schema	SaleLine.xsd

NameSpace	http://www.BikeOnTrack.pt/SaleLine
Descrição	<p>Temos presente um ComplexType com 5 elementos: id, produtos, unitPrice, order, lineTotal.</p> <p>O elemento produtos é ele mesmo um ComplexType, contendo dentro de si um outro elemento produtoType, que é do tipo p:produto , este tipo foi importado através do schema http://www.BikeOnTrack.pt/Produto, sendo este condicionado a existir uma única vez.</p>
Elementos filho	sl:id, sl:produtos, sl:unitPrice, sl:order, sl:linetotal
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/SaleLine" targetNamespace="http://www.BikeOnTrack.pt/SaleLine" xmlns:ct="http://www.BikeOnTrack.pt/CommonTypes" xmlns:p="http://www.BikeOnTrack.pt/Produto" elementFormDefault="qualified"> <xs:import schemaLocation="CommonTypes.xsd" namespace="http://www.BikeOnTrack.pt/CommonTypes"/> <xs:import schemaLocation="Produto.xsd" namespace="http://www.BikeOnTrack.pt/Produto"/> <xs:element name="saleline" > <xs:complexType> <xs:sequence> <xs:element name="id" type="ct:id"/> <xs:element name="produtos"> <xs:complexType> <xs:sequence> <xs:element name="produtoType" type="p:produto" minOccurs="1" maxOccurs="1"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="unitPrice" type="ct:valor"/> <xs:element name="order" type="ct:id"/> <xs:element name="lineTotal" type="ct:valor"/> </xs:sequence> </xs:complexType> </xs:element> </xs:schema> </pre>
Dependências em relação a outros elementos (identificar schema)	S:Sales

NameSpace	http://www.BikeOnTrack.pt/Produto
Propriedades	elementFormDefault="qualified"
Nome do Schema	Produto.xsd

NameSpace	http://www.BikeOnTrack.pt/Produto
Descrição	Temos presente um ComplexType com 5 elementos: id, name, productNumber, color e price. Os quais o type é especificado no xsd CommonTypes.
Elementos filho	p:id, p:name, p:productNumber, p:color, p:price
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/Produto" targetNamespace="http://www.BikeOnTrack.pt/Produto" xmlns:ct="http://www.BikeOnTrack.pt/CommonTypes" elementFormDefault="qualified"> <xs:import schemaLocation="CommonTypes.xsd" namespace="http://www.BikeOnTrack.pt/ CommonTypes"/> <xs:complexType name="produto"> <xs:sequence> <xs:element name="id" type="ct:id"/> <xs:element name="name" type="ct:texto"/> <xs:element name="productNumber" type="ct:productNum"/> <xs:element name="color" type="ct:texto"/> <xs:element name="price" type="ct:valor"/> </xs:sequence> </xs:complexType> </xs:schema> </pre>
Dependências em relação a outros elementos (identificar schema)	sl:SaleLine

NameSpace	http://www.BikeOnTrack.pt/CurrencyDetails
Propriedades	elementFormDefault="qualified"
Nome do Schema	CurrencyDetails.xsd

NameSpace	http://www.BikeOnTrack.pt/CurrencyDetails
Descrição	Temos presente um ComplexType com 5 elementos: id, currencyRateDate, fromCurrencyCode, toCurrencyCode e rateVal; Os quais o type é especificado no xsd CommonTypes.
Elementos filho	cd:id, cd:currencyRateDate, cd:fromCurrencyCode, cd:toCurrencyCode e cd:rateVal;
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/CurrencyDetails" targetNamespace="http://www.BikeOnTrack.pt/CurrencyDetails" xmlns:ct="http://www.BikeOnTrack.pt/CommonTypes" elementFormDefault="qualified"> <xs:import schemaLocation="CommonTypes.xsd" namespace="http://www.BikeOnTrack.pt/CommonTypes"/> <xs:complexType name="currencydetails"> <xs:sequence> <xs:element name="id" type="ct:id"/> <xs:element name="currencyRateDate" type="xs:dateTime"/> <xs:element name="fromCurrencyCode" type="ct:code"/> <xs:element name="toCurrencyCode" type="ct:code"/> <xs:element name="rateVal" type="ct:valor"/> </xs:sequence> </xs:complexType> </xs:schema> </pre>
Dependências em relação a outros elementos (identificar schema)	s:Sales

NameSpace	http://www.BikeOnTrack.pt/ReceiptInfo
Propriedades	elementFormDefault="qualified"
Nome do Schema	ReceiptInfo.xsd

NameSpace	http://www.BikeOnTrack.pt/ReceiptInfo
Descrição	Temos presente um ComplexType com 2 elementos: dataInicio, dataFim. Os quais o type é especificado no xsd CommonTypes.
Elementos filho	ri:dataInicio, ri:DataFim
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/ReceiptInfo" targetNamespace="http://www.BikeOnTrack.pt/ReceiptInfo" elementFormDefault="qualified"> <xs:complexType name="info"> <xs:sequence> <xs:element name="dataInicio" type="xs:dateTime"/> <xs:element name="dataFim" type="xs:dateTime"/> </xs:sequence> </xs:complexType> </xs:schema> </pre>
Dependências em relação a outros elementos (identificar schema)	Exercise

NameSpace	http://www.BikeOnTrack.pt/CommonTypes
Propriedades	elementFormDefault="qualified"
Nome do Schema	CommonTypes.xsd

NameSpace	http://www.BikeOnTrack.pt/CommonTypes
Descrição	Neste xsd especificamos as restrições aos dados que encontramos na pesquisa do problema, desta forma estes agrupam os dados que partilham as mesmas características.
Elementos filho	ct:valor, ct:texto, ct:morada, ct:code, ct:id, ct:number, ct:productNum, ct:telefone, ct:nif
Estrutura XSD	<pre> <?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.BikeOnTrack.pt/CommonTypes" targetNamespace="http://www.BikeOnTrack.pt/CommonTypes" elementFormDefault="qualified"> <xs:simpleType name="valor"> <xs:restriction base="xs:decimal"> <xs:minInclusive value="0"/> <xs:totalDigits value="15"/> <xs:fractionDigits value="4"/> </xs:restriction> </xs:simpleType> <xs:simpleType name="texto"> <xs:restriction base="xs:string"> <xs:pattern value="([a-zA-Z 0-9, -])*/> </xs:restriction> </xs:simpleType> <xs:simpleType name="morada"> <xs:restriction base="xs:string"> <xs:pattern value="([a-zA-Z0-9 -])*/> </xs:restriction> </xs:simpleType> <xs:simpleType name="code"> <xs:restriction base="xs:string"> <xs:pattern value="[A-Z]{3}"/> </xs:restriction> </xs:simpleType> </pre>

Estrutura XSD	<pre> <xs:simpleType name="number"> <xs:restriction base="xs:integer"> <xs:minInclusive value="0"/> <xs:totalDigits value="15"/> <xs:fractionDigits value="0"/> </xs:restriction> </xs:simpleType> <xs:simpleType name="productNum"> <xs:restriction base="xs:string"> <xs:pattern value="([a-zA-Z0-9\-\])*"/> </xs:restriction> </xs:simpleType> <xs:simpleType name="telefone"> <xs:restriction base="xs:string"> <xs:pattern value="([0-9+])*"/> </xs:restriction> </xs:simpleType> <xs:simpleType name="nif"> <xs:restriction base="xs:string"> <xs:pattern value="[0-9]{9}"/> </xs:restriction> </xs:simpleType> </xs:schema> </pre>
Dependências em relação a outros elementos (identificar schema)	

Nota: *elementFormDefault= "qualified"* — Utilizamos o "qualified" em todos os XSDs para que a leitura do XML tenha o prefixo do XSD, de forma a ver de onde surge o elemento e de que natureza é.

Exemplo XML

```
<?xml version="1.0" encoding="utf-8"?>

<exerciseType xmlns="http://www.BikeOnTrack.pt/Exercise"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.BikeOnTrack.pt/Exercise Exercise.xsd"
  xmlns:c="http://www.BikeOnTrack.pt/Cliente"
  xmlns:l="http://www.BikeOnTrack.pt/Loja"
  xmlns:p="http://www.BikeOnTrack.pt/Produto"
  xmlns:ri="http://www.BikeOnTrack.pt/ReceiptInfo"
  xmlns:cd="http://www.BikeOnTrack.pt/CurrencyDetails"
  xmlns:s="http://www.BikeOnTrack.pt/Sales"
  xmlns:sl="http://www.BikeOnTrack.pt/SaleLine"
  >
  <salesTotal>
    <s:saleType>
      <s:id>43659</s:id>
      <s:lojaInfo>
        <s:lojaType>
          <l:id>1046</l:id>
          <l:name>Better Bike Shop</l:name>
          <l:nif>267315228</l:nif>
          <l:morada>rua do monte</l:morada>
          <l:contacto>910052980</l:contacto>
        </s:lojaType>
      </s:lojaInfo>
      <s:clientInfo>
        <s:clientType>
          <c:id>29825</c:id>
          <c:nome>Pedro</c:nome>
          <c:nif>241011647</c:nif>
        </s:clientType>
      </s:clientInfo>
      <s:orderDate>2011-05-31T00:00:00.000</s:orderDate>
      <s:line>
        <sl:saleline>
          <sl:id>1</sl:id>
          <sl:produtos>
            <sl:produtoType>
              <p:id>776</p:id>
              <p:name>Mountain-100 Black, 42</p:name>
              <p:productNumber>BK-M82B-42</p:productNumber>
              <p:color>Black</p:color>
              <p:price>3374.99</p:price>
            </sl:produtoType>
          </sl:produtos>
          <sl:unitPrice>2024.994</sl:unitPrice>
          <sl:order>1</sl:order>
          <sl:lineTotal>2024.994</sl:lineTotal>
        </sl:saleline>
      </s:line>
    </s:saleType>
  </salesTotal>
</exerciseType>
```

```

        <sl:saleline>
          <sl:id>1</sl:id>
          <sl:produtos>
            <sl:produtoType>
              <p:id>776</p:id>
              <p:name>Mountain-100 Black, 42</p:name>
              <p:productNumber>BK-M82B-42</p:productNumber>
              <p:color>Black</p:color>
              <p:price>3374.99</p:price>
            </sl:produtoType>
          </sl:produtos>
          <sl:unitPrice>2024.994</sl:unitPrice>
          <sl:order>1</sl:order>
          <sl:lineTotal>2024.994</sl:lineTotal>
        </sl:saleline>
      </s:line>
      <s:currency>
      </s:currency>
      <s:taxAmt>1971.5149</s:taxAmt>
      <s:subTotal>20565.6206</s:subTotal>
      <s:total>22537.1355</s:total>
    </s:saleType>
    <s:saleType>
      <s:id>43659</s:id>
      <s:lojaInfo>
        <s:lojaType>
          <l:id>1046</l:id>
          <l:name>Better Bike Shop</l:name>
          <l:nif>267315228</l:nif>
          <l:morada>rua do monte</l:morada>
          <l:contacto>910052980</l:contacto>
        </s:lojaType>
      </s:lojaInfo>
      <s:clientInfo>
        <s:clientType>
          <c:id>29825</c:id>
          <c:nome>Pedro</c:nome>
          <c:nif>241011647</c:nif>
        </s:clientType>
      </s:clientInfo>
      <s:orderDate>2011-05-31T00:00:00.000</s:orderDate>
      <s:line>
        <sl:saleline>
          <sl:id>1</sl:id>
          <sl:produtos>
            <sl:produtoType>
              <p:id>776</p:id>
              <p:name>Mountain-100 Black, 42</p:name>
              <p:productNumber>BK-M82B-42</p:productNumber>
              <p:color>Black</p:color>
              <p:price>3374.99</p:price>
            </sl:produtoType>

```



```

        </sl:produtos>
        <sl:unitPrice>2024.994</sl:unitPrice>
        <sl:order>1</sl:order>
        <sl:lineTotal>2024.994</sl:lineTotal>
    </sl:saleline>
    <sl:saleline>
        <sl:id>1</sl:id>
        <sl:produtos>
            <sl:produtoType>
                <p:id>776</p:id>
                <p:name>Mountain-100 Black, 42</p:name>
                <p:productNumber>BK-M82B-42</p:productNumber>
                <p:color>Black</p:color>
                <p:price>3374.99</p:price>
            </sl:produtoType>
        </sl:produtos>
        <sl:unitPrice>2024.994</sl:unitPrice>
        <sl:order>1</sl:order>
        <sl:lineTotal>2024.994</sl:lineTotal>
    </sl:saleline>
</s:line>
<s:currency>
    <s:currencyType>
        <cd:id>10</cd:id>
        <cd:currencyRateDate>2011-05-31T00:00:00.000</cd:currencyRateDate>
        <cd:fromCurrencyCode>USD</cd:fromCurrencyCode>
        <cd:toCurrencyCode>JPY</cd:toCurrencyCode>
        <cd:rateVal>104.958</cd:rateVal>
    </s:currencyType>
</s:currency>
<s:taxAmt>1971.5149</s:taxAmt>
<s:subTotal>20565.6206</s:subTotal>
<s:total>22537.1355</s:total>
</s:saleType>
</salesTotal>
<information>
    <infoType>
        <ri:dataInicio>2011-05-31T00:00:00.000</ri:dataInicio>
        <ri:dataFim>2011-08-31T00:00:00.000</ri:dataFim>
    </infoType>
</information>
</exerciseType>

```

Conclusão

Neste trabalho abordámos conceitos de anotação de documentos e as tecnologias correntes para a especificação de linguagens de anotação, como também demonstramos conhecimento da semântica e da sintaxe da linguagem XML, e concluímos que esta linguagem é uma grande vantagem, derivado do facto que pode ser usado para armazenamento de informação em qualquer plataforma, por ter um padrão internacional, como para comunicação e transferência de dados entre diferentes sistemas computacionais que seriam incapazes de se comunicar, sem precisar se preocupar com tipo de dado.

Cumprimos todos os objetivos que nos tínhamos proposto, que era desenvolver um vocabulário XML para facilitar a comunicação entre os sistemas informáticos das lojas da BikeOnTrack.

Para concluir, a forma como desenvolvemos o vocabulário tem as suas vantagens, como por exemplo, é fácil de entender o processo todo, mesmo para uma pessoa que não perceba da linguagem e é fácil a alteração dos documentos caso seja necessário. Como também tem as suas desvantagens, como é o caso do “ref”, em que a desvantagem de usar este método é o facto de que passa a ser um elemento root, logo pode ser criado como um XML isolado.

Referências

<https://moodle.estg.ipp.pt/>