

Gestor de Vendas de uma Padaria

Trabalho Prático no âmbito da cadeira de Engenharia de Dados e Conhecimento

Francisco Cunha, nº 76759

João Amaral, nº 76460

João Maia, nº 76364

1. Introdução ao Tema

O tema escolhido para o nosso trabalho trata da criação de uma plataforma online que permita um administrador de um negócio (no nosso caso de uma padaria) realizar a sua gestão, ou seja, gerir os diversos produtos que vende, assim como as suas vendas e clientes associados. Através desta plataforma este pode pesquisar por produtos, clientes ou transações, obter informações detalhadas de cada um destes, assim como inserir novos clientes, produtos, entre outras operações.

2. Dados e suas fontes

Os dados usados neste projeto foram, numa primeira fase, obtidos através de um documento no formato XML (*eXtensible Markup Language*) criado automaticamente por um programa de certificação de vendas de uma padaria real. No entanto devido à sua difícil compreensão e complexidade, quer ao nível estrutural quer ao nível lógico, optámos pela criação de um novo ficheiro XML (*bakery_sales.xml*), contendo toda a informação útil do ficheiro original, mas possuindo uma melhor organização dos dados, isto é, elementos bem definidos seguindo uma estrutura em árvore e com um nome bem definido e explícito.

Por ventura, este ficheiro foi importado para uma base de dados nativa em xml, o BaseX, a qual é usada para as operações disponíveis pela plataforma, através de queries na linguagem de programação funcional Xquery.

3. Esquema dos dados (XML Schema)

O ficheiro XML possui 1 elemento raíz (*document node*) designado de “*CompanySales*” com 4 filhos sendo todos elementos complexos por serem compostos por vários elementos: “*Company*”, “*CompanyProducts*”, “*CompanyClients*” e “*CompanyTransactions*”.

```
<CompanySales>
  <Company>...</Company>
  <CompanyProducts>...</CompanyProducts>
  <CompanyClients>...</CompanyClients>
  <ClientsTransactions>...</ClientsTransactions>
</CompanySales>
```

O elemento “*Company*”, que armazena informação da companhia proprietária da aplicação, possui 8 filhos, sendo 6 elementos simples e os 2 restantes elementos complexos:

```
<Company>
  <CompanyId>...</CompanyId>
  <TaxRegistrationNumber>...</TaxRegistrationNumber>
  <CompanyName>...</CompanyName>
  <BusinessName>...</BusinessName>
  <CurrencyCode>...</CurrencyCode>
  <Address>
    <BuildingNumber>...</BuildingNumber>
    <StreetName>...</StreetName>
    <AddressDetail>...</AddressDetail>
    <City>...</City>
    <PostalCode>...</PostalCode>
    <Region>...</Region>
    <Country>...</Country>
  </Address>
  <Contact>
    <Telephone>...</Telephone>
    <Fax>...</Fax>
    <Email>...</Email>
    <Website>...</Website>
  </Contact>
</Company>
```

- “*CompanyId*”: número de Identidade da Companhia;
- “*TaxRegistrationNumber*”: número Fiscal da Companhia;
- “*CompanyName*”: nome da Companhia;
- “*BusinessName*”: nome do estabelecimento onde se encontra o sistema com a aplicação;
- “*CurrencyCode*”: código da moeda usada no estabelecimento;
- “*Address*”: este é um elemento complexo, já que é composto por 7 elementos simples, seus filhos, e guarda a descrição do endereço do estabelecimento:
 - “*BuildingNumber*”: número do prédio onde se localiza o estabelecimento;
 - “*StreetName*”: nome da rua onde se encontra o estabelecimento;
 - “*AddressDetail*”: tal como “*StreetName*” complementa mais a informação da morada
 - “*City*”: cidade onde se encontra o estabelecimento;
 - “*PostalCode*”: código postal do estabelecimento;
 - “*Region*”: região a qual pertence o estabelecimento;
 - “*Country*”: país onde se localiza o estabelecimento;
- “*Contact*”: este é um elemento complexo por ser composto por 4 elementos simples, seus filhos:
 - “*Telephone*”: número de telefone do estabelecimento;
 - “*Fax*”: fax do estabelecimento;
 - “*Email*”: endereço de *email* do estabelecimento;
 - “*Website*”: *website* do estabelecimento;

O elemento “*CompanyProducts*” armazena informação dos produtos disponíveis no stock do estabelecimento e é composto por vários elementos complexos, os elementos “*Product*”. Estes elementos são compostos por 4 filhos, sendo 3 elementos simples e o restante um elemento complexo:

```
<CompanyProducts>
  <Product>
    <ProductID>...</ProductID>
    <ProductName>...</ProductName>
    <UnitCost>...</UnitCost>
    <Tax>
      <TaxType>...</TaxType>
      <Description>...</Description>
      <TaxPercentage>...</TaxPercentage>
    </Tax>
  </Product>
  ....
```

</CompanyProducts>

- “*ProductID*”: número de identidade do produto;
- “*ProductName*”: nome do produto;
- “*UniCost*”: custo por unidade, em euros;
- “*Tax*”: este elemento complexo com finalidade de armazenar informação da taxa do produto é composto por 4 elementos simples:
 - “*TaxType*”: tipo de taxa a pagar;
 - “*Description*”: descrição da taxa a pagar;
 - “*TaxPercentage*”: percentagem da taxa a pagar;

O elemento complexo “*CompanyClients*” guarda a informação dos clientes do estabelecimento e é constituído por vários elementos complexos denominados “*Clients*”. Cada elemento “*Client*” é constituído por 4 filhos, 2 desses filhos sendo elementos simples e os restantes 2 sendo elementos complexos:

```
<CompanyClients>
  <Client>
    <ClientID>...</ClientID>
    <ClientName>...</ClientName>
    <ClientAddress>
      <AddressDetail>...</AddressDetail>
      <City>...</City>
      <PostalCode>...</PostalCode>
      <Region>...</Region>
      <Country>...</Country>
    </ClientAddress>
    <ClientContact>
      <Telephone>...</Telephone>
      <Fax>...</Fax>
      <Email>...</Email>
    </ClientContact>
  </Client>
  ...
</CompanyClients>
```

- “*ClientID*”: número de identidade do cliente;
- “*ClientName*”: nome do cliente;
- “*ClientAddress*”: este elemento complexo com finalidade de guardar informação do endereço do cliente é composto por 5 elementos simples:
 - “*AddressDetail*”: morada do cliente;

- “*City*”: cidade onde mora o cliente;
- “*PostalCode*”: código postal do endereço do cliente;
- “*Region*”: região onde o cliente reside;
- “*Country*”: país onde o cliente reside;
- “*ClientContract*”: este elemento complexo com o objetivo de guardar informação do contacto do cliente é composto por 3 elementos simples:
 - “*Telephone*”: número de telefone do cliente;
 - “*Fax*”: fax do cliente;
 - “*Email*”: endereço de *email* do cliente;

Finalizando os elementos filhos da raiz, o elemento complexo “*ClientsTransactions*” arquiva a informação das vendas efetuadas pelos clientes e é formado por vários elementos complexos “*Transaction*”. Cada um destes elementos é constituído por 8 filhos, sendo 7 elementos simples e o restante um elemento complexo:

```
<ClientsTransactions>
  <Transaction>
    <TransactionID>...</TransactionID>
    <ClientID>... </ClientID>
    <ClientName>...</ClientName>
    <ProductID>...</ProductID>
    <ProductName>...</ProductName>
    <ProductQuantity>...</ProductQuantity>
    <TotalCost>...</TotalCost>
    <TransactionTime>
      <Date>...</Date>
      <Time>...</Time>
    </TransactionTime>
  </Transaction>
  ...
</ClientsTransactions>
```

- “*TransactionID*”: número de identificação da venda efetuada;
- “*ClientID*”: número de identificação do cliente que efetuou a venda;
- “*ClientName*”: nome do cliente que efetuou a venda;
- “*ProductID*”: número de identificação do produto vendido;
- “*ProductName*”: nome do produto vendido;
- “*ProductQuantity*”: número de unidades vendidas;
- “*TotalCost*”: custo total da venda efetuada;

- “*TransactionTime*”: elemento complexo, que armazena a informação do horário em que a venda foi efetuada, formado por 2 elementos simples:
 - “*Date*”: data da venda efetuada;
 - “*Time*”: hora da venda efetuada;

Por fim, o ficheiro XML Schema foi utilizado para validação do ficheiro XML e para visualização do esquema da árvore produzida no ficheiro XML, sendo que todos os dados armazenados nos elementos são do tipo *string*.

4. Transformações sobre os dados (XSLT)

A aplicação possui um ficheiro XSL para transformação de um ficheiro XML que contém informação da meteorologia da região de Aveiro, retirada do *website* “www.bbc.com/weather/2742611”, ou seja, um *RSS Feed*, num documento HTML usado em conjunto com o documento HTML “*index.html*” para reproduzir a página inicial da aplicação.

5. Operações sobre os dados (XQuery e XQuery Update)

As operações implementadas na aplicação são:

- **Ordenação de resultados**

Permite a ordenação dos resultados obtidos de forma ascendente ou descendente, quer seja de ordenação de produtos, clientes ou transações.

- **Pesquisa de dados**

Permite pesquisar dados, sejam esses dados tanto de clientes ou de produtos.

- **Adicionar clientes**

Permite adicionar um novo cliente, registando os campos e valores necessários, fazendo depois a actualização dos dados.

- **Listagem de clientes**

Permite fazer a listagem dos clientes existentes.

- **Informação de clientes**

Dado um cliente, é mostrado ao utilizador os dados referentes a esse mesmo cliente.

- **Alterar informação de clientes**

Dado um determinado cliente, é possível alterar a informação relativa a esse cliente.

- **Adicionar produtos**

Permite adicionar um novo produto, registando os campos e valores necessários, fazendo depois a actualização dos dados.

- **Listagem de produtos**

Permite fazer a listagem dos produtos existentes.

- **Informação de produtos**

Dado um determinado produto, é possível ver a informação relativa a esse determinado produto.

- **Alterar informação de produtos**

Dado um determinado produto, é possível alterar a informação relativa a esse produto.

- **Listagem de IVA**

Permite verificar os diversos tipos de IVA existentes.

- **Informação relativa ao IVA**

Permite verificar que produtos existem com uma determinada taxa de IVA.

- **Adicionar vendas**

Permite adicionar uma nova venda, registando os campos e valores necessários, fazendo depois a actualização dos dados.

- **Listagem de vendas**

Permite verificar as diversas vendas efectuadas.

- **Informação de vendas**

Dada uma determinada venda, é possível saber qual foi o cliente para o qual a transação foi feita, os produtos vendidos, entre outras informações.

6. Funcionalidades da aplicação

As funcionalidades da aplicação são as seguintes:

- **Redirecionar para a página principal**

Permite redireccionar para a página principal, independentemente da página onde o utilizador está.

- **Pesquisar por produtos ou clientes**

O utilizador pesquisa por um determinado produto ou cliente, e se existir, então o correspondente resultado da pesquisa é mostrado.

- **Adicionar clientes**

Permite ao utilizador adicionar novos clientes, com base em campos específicos que o utilizador deve preencher de forma a conseguir adicionar com sucesso novos clientes.

- **Listar clientes**

O utilizador quando lista os clientes, é-lhe listada uma tabela com os clientes existentes, podendo o utilizador carregar no nome de um cliente de forma a poder ver as informações desse determinado cliente.

- **Ver informações de clientes**

É mostrado ao utilizador as diversas informações de um determinado cliente, como o identificador do cliente, a morada, contacto, país de origem, entre outros.

- **Alterar informações de clientes**

O utilizador pode alterar as informações de um determinado cliente, como por exemplo o nome do cliente, a morada, entre outros.

- **Adicionar produtos**

Tal como o utilizador pode adicionar novos clientes, também é possível adicionar novos produtos, segundo campos que lhe são solicitados.

- **Listar produtos**

Faz uma listagem ao utilizador dos produtos existentes e se o utilizador carregar num determinado produto é-lhe mostrada as informações desse mesmo produto.

- **Ver informações de produtos**

É mostrado ao utilizador as informações relativas a esse produto, como por exemplo o seu preço, o seu IVA e a sua taxa. Se o utilizador carregar

- **Alterar informações de produtos**

O utilizador pode alterar as informações de um determinado produto, como por exemplo o nome do produto, o preço, entre outros.

- **Ver tipos de IVA**

É mostrada a informação relativa dos diversos tipos de IVA ao utilizador.

- **Verificar produtos com determinada taxa de IVA**

O utilizador pode verificar quais os produtos existentes com uma determinada taxa de IVA aplicada.

- **Adicionar venda**

O utilizador pode adicionar uma nova venda, tendo apenas que preencher os campos necessários de forma a conseguir registar uma venda com sucesso.

- **Listar vendas**

É mostrado ao utilizador as diversas vendas efectuadas aos vários clientes. Se o utilizador carregar num determinado cliente, é-lhe mostrada as várias vendas efetuadas para esse mesmo cliente.

- **Verificar vendas**

É permitido ao utilizador verificar a venda efectuada para um determinado cliente, e as informações que lhe são mostradas são por exemplo, o identificador do produto vendido, a quantidade, o preço total, a data e hora.

7. Conclusões

Neste projecto, foi abordado a criação de uma plataforma online que permita um administrador de um negócio (neste caso de uma padaria) realizar a sua gestão.

Tentou-se realizar todos os objetivos propostos da melhor forma possível, havendo ainda a possibilidade de adicionar mais funções na aplicação e torná-la mais eficiente.

Com este projecto, os elementos do grupo desenvolveram e aprimoraram as capacidades com novas tecnologias que não tinham experienciado antes, permitindo aos elementos do grupo explorar as suas funcionalidades e adquirir experiência nessas tecnologias.

8. Pré-requisitos da aplicação

Para executar a aplicação é necessário ter instalado no sistema os seguintes módulos:

- a. Python 3.6
- b. Requests module
 - i. `> pip3 install requests`
- c. LXML module
 - i. `> pip install lxml`

9. Configuração para executar a aplicação

As seguintes configurações são necessárias para a correta execução da aplicação:

1. Num terminal executar o comando `java -cp path/to/BaseX867.jar org.basex.BaseXServer`;
2. Num outro terminal correr o comando `java -jar path/to/BaseX867.jar`
3. Na GUI que abriu, carregar em "New", e seguidamente adicionar o ficheiro "bakery_sales.xml";
4. No PyCharm, correr a aplicação ou executar o comando na pasta da aplicação `python3 path/to/manage.py runserver`;