

# 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 este trabalho é o mesmo que o trabalho prático anterior e trata-se 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, suas fontes e a sua transformação

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.

Através de um ficheiro XSL (*xml\_to\_n3.xsl*), os dados provenientes do ficheiro XML foram transformados num ficheiro RDF no formato n3. Este formato foi escolhido por apresentar simplicidade na estrutura dos dados devido à utilização dos prefixos e uma apresentação intuitiva e acessível.

### 3. Operações sobre os dados

Muitas das operações implementadas na aplicação são provenientes do primeiro trabalho prático, mudando as pesquisas efectuadas para o novo modelo de dados. As novas funcionalidades são “**Informações adicionais sobre cidades dos clientes**” e “**Informações adicionais sobre determinados produtos**”. Todas as funcionalidades implementadas encontram-se descritas abaixo:

- **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.

- **Informações adicionais sobre cidades dos clientes (nova)**

Dado uma determinada cidade, é possível visualizar determinadas informações relativas a essa cidade.

- **Adicionar produtos**

Permite adicionar um novo produto, preenchendo 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.

- **Informações adicionais sobre determinados produtos (nova)**

Dado um determinado produto, é possível visualizar certas informações relativas 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.

## 4. A publicação de dados semânticos através de RDFa

Para a publicação da semântica de dados na aplicação foram usadas alguns *schemas* obtidos do website “*schemas.org*”. Esta informação está presente nas páginas onde é apresentado ao utilizador a listagem de produtos, clientes ou transações, assim como na página de pesquisa, mais concretamente nos resultados obtidos da realização da mesma.

Na semântica de dados de um cliente do nosso sistema foi usada a schema “*Place*”, uma vez que é aquela que melhor representa o significado de um cliente no neste contexto. Nesta mesma foram usadas as propriedades (*properties*) de “*Name*”, “*Identifier*” (identificador de cada utilizador), “*Address*” (endereço), “*Telephone*”, “*faxNumber*” e “*Email*”. Foi ainda usada uma schema adicional (*openvocab.org*) para representar o país (“*country*”) e código-postal (“*postalAddress*”), uma vez que “*Place*” não possui nenhuma das duas.

Para os produtos foi usada a schema “*Product*” uma vez que definia concretamente o contexto de um produto no nosso sistema. Desta foram usadas as propriedades de “*Name*”, assim como uma outra schema, “*PriceSpecification*”, para representar o valor monetário de cada produto, mais especificamente usando as propriedades “*valueAddedTaxIncluded*”, “*Price*” e “*Description*”.

Na representação semântica de uma transação foi usada a *schema* “*MoneyTransfer*” da qual foram usadas as propriedades “*Identifier*”, para exprimir um identificador de transação, “*purchaseDate*” para a data desta, e ainda “*Agent*” para expressar o cliente que realizou dada transação. Adicionalmente foi usada a *schema* “*priceSpecification*” juntamente com a propriedade “*price*” para representar o valor monetário da transação.

Uma das dificuldades encontradas durante a realização desta secção do trabalho foi a escolha das *schemas* que melhor caracterizassem semanticamente os nossos clientes, produtos e transações. Deste modo optámos por escolher algumas *schemas* que embora possuam um menor número de “*properties*” para o nosso contexto específico (por exemplo “*Product*”), eram mais concretas no seu significado, (por exemplo escolher “*Product*” para representar um produto em vez de “*Thing*”).

## 5. A integração de dados da Wikidata

As integrações de dados da Wikidata são as seguintes:

- **Ver informações adicionais de determinados produtos**

É possível visualizar informações de determinados produtos. Consoante os produtos seleccionados é possível visualizar informações como o país de origem desse produto, alguns dos seus ingredientes, entre outros e uma imagem desse produto.

- **Ver informações de cidades dos clientes**

É possível também visualizar informações das cidades dos clientes. Consoante a cidade do cliente, é possível visualizar informações dessa cidade tal como a área da cidade, uma imagem de um monumento e a quantidade de pessoas residentes dessa cidade.

## 6. Funcionalidades da aplicação

A maioria das funcionalidades da aplicação foram mantidas do trabalho prático anterior, no entanto algumas novas funcionalidades foram inseridas. As funcionalidades 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 resultado da pesquisa correspondente 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.

- **Informações adicionais sobre cidades dos clientes (nova)**

O utilizador pode visualizar informações de uma determinada cidade dos clientes, tais como a área da cidade e o número de habitantes.

- **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.

- **Informações adicionais sobre determinados produtos (nova)**

É permitido ao utilizador visualizar informações de determinados produtos como por exemplo o seu país de origem e alguns dos seus ingredientes.

- **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 efectuadas 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

Este projeto utilizou as bases do trabalho prático anterior, sendo que maior parte das funcionalidades e objetivos se mantiveram. A diferença significativa foi a mudança das tecnologias e dos métodos utilizados.

Na opinião do grupo, a linguagem *SPARQL* é mais intuitiva e simples de utilizar em relação à linguagem de *queries* utilizada previamente, o *XQuery*, principalmente devido à utilização dos prefixos que reduzem a quantidade e a complexidade do código e tornam mais intuitivo a utilização dos *URIs*, e a organização e a estrutura das queries que simplificam a escrita da *query* e deteção de erros nestas.

Considerou-se a biblioteca *s4api* como sendo útil e bastante acessível como ferramenta de conexão entre o código *python* e a base de dados *GraphDB*, contudo, um ponto fraco desta biblioteca é a distinção das *queries* de *SELECT* e *UPDATE*, sendo necessário recorrer a funções distintas para cada uma.

A base de dados *GraphDB* foi apontada como uma boa ferramenta para testar *queries* e manipular os dados de um repositório localizado na base de dados, porém, a remoção de todos os dados de um ficheiros *rdf*, ou seja, a limpeza total do repositório, foi visto como uma falha, já que a opção de limpeza de todos os dados contidos no repositório se encontra escondida e não num local intuitivo.

Por fim, o grupo aumentou as capacidades de alteração e redefinição da estrutura de um projeto considerado como completo e a transição de uma tecnologia familiarizada para uma tecnologia nova e não familiarizada.

## 8. Configuração para executar a aplicação

As seguintes configurações são necessárias para a correta execução da aplicação, em Linux / OS X:

1. Inicializar o *graphDB* e criar um repositório com o nome “ProjEDC2”
2. Fazer o *upload* do ficheiro “*bakery\_data.n3*”, localizado no diretório “*/bills/templates/*”, para o repositório criado.
3. No PyCharm, correr a aplicação ou executar o comando na pasta da aplicação “`python3 path/to/manage.py runserver`”;

## 9. Pré-requisitos da aplicação

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

- a. Python 3.6
- b. s4Api
  - i. `> pip3 install s4api`
- c. SparqlWrapper
  - i. `> pip3 install SparqlWrapper`
- d. Requests module
  - i. `> pip3 install requests`
- e. LXML module
  - i. `> pip install lxml`
- f. Pandas module
  - i. `> pip3 install pandas`