

UNIVERSIDADE DO MINHO
MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
PROCESSAMENTO DE LINGUAGENS E CONHECIMENTO

Processamento e Representação de Conhecimento

Projeto

Gonçalo Pinto, A83732
Luís Ribeiro - A85954

28 de junho de 2021

Resumo

O presente trabalho foi desenvolvido no âmbito da unidade curricular Processamento e Representação de Conhecimento do 2º semestre, do 1º ano do Mestrado em Engenharia Informática da Universidade do Minho, esta tem como objetivo a aprendizagem prática de conceitos desenvolvidos associados ao perfil de Processamento de Linguagens e Conhecimento. O presente trabalho consistiu na especificação de conhecimento de forma a conseguir construir uma aplicação informática que tire partido desse conhecimento.

Conteúdo

1	Introdução	3
2	Contextualização e Trabalhos Relacionados	4
2.1	Contextualização	4
2.2	Trabalhos Relacionados	4
3	Metodologia, Tecnologias Utilizadas e Dataset	5
3.1	Metodologia	5
3.2	Tecnologias Utilizadas	5
3.3	Dataset	6
4	Implementação	7
4.1	Ontologia	7
4.2	Tratamento dos Dados	9
4.3	Aplicação	10
5	Conclusão e Trabalho Futuro	17

Lista de Figuras

4.1	Estrutura da ontologia, imagem obtida através de <i>WebVOWL</i> .	9
4.2	Página inicial da aplicação.	12
4.3	Resultado de uma pesquisa sobre um determinada bebida e sobre um determinado <i>bartender</i> .	12
4.4	Lista de indivíduos de uma classe presente na ontologia.	13
4.5	Indivíduo de uma classe presente na ontologia.	13
4.6	Lista de <i>Cocktails</i> .	14
4.7	Página individual de um <i>Cocktail</i> .	14
4.8	Combinar ingredientes.	14
4.9	Gestão de conhecimento.	15
4.10	Gestão dos elementos da classe <i>Cocktail</i> .	15
4.11	Gestão de um elemento da classe <i>Cocktail</i> .	15
4.12	Comentários introduzidos pelos utilizadores.	16
4.13	Locais visitados pelos utilizadores.	16
4.14	Perfil de um utilizador.	16

Capítulo 1

Introdução

No 2º semestre, do 1º ano do Mestrado em Engenharia Informática da Universidade do Minho, existe uma unidade curricular enquadrada no perfil de Processamento de Linguagens e Conhecimento denominada por Processamento e Representação de Conhecimento, que tem como objetivo agilizar a capacidade dos alunos no processo de especificação formal usando várias metodologias, tais como, taxonomias, *thesaurus* e ontologia e no processo de adição semântica descritiva a objetos digitais, desenvolver a capacidade de especificar ontologias em OWL, bem como, potenciar a habilidade de processar ontologias e a competência de adicionar semântica a sítios web usando as normas *RDFa* e "*Open Linked Data*".

O presente trabalho enquadra-se nesta unidade curricular e pretendeu-se explorar conhecimento de forma a conseguir construir uma aplicação informática que tire partido do mesmo. Para tal, foi necessário escolher um tema e um conjunto de dados relativo ao mesmo, de forma a desenvolver uma interface web interativa que disponibilize a informação de forma fácil e acessível.

Neste relatório irá ser detalhada a contextualização e trabalhos relacionados com o pretendido; qual a metodologia, tecnologias utilizadas e o conjunto de dados, *dataset*, utilizado; o processo de implementação da aplicação e decisões relativas ao desenvolvimento do mesmo. Para além disto, será feita também uma apreciação do sistema desenvolvido.

Capítulo 2

Contextualização e Trabalhos Relacionados

2.1 Contextualização

Desde sempre que as pessoas consomem álcool, como também misturaram várias bebidas numa só adicionando ao mesmo um toque de tempero. Por exemplo, os gregos temperavam o seu vinho com tudo, desde mel até água do mar. É justo dizer que o desejo de aumentar os efeitos da bebida e de melhorar o sabor de um determinado produto intensificando ou diminuindo o seu efeito tem estado presente na raça humana há muito tempo.[1]

A mistura de uma ou mais bebidas alcoólicas, denominada de *cocktail*, é uma combinação de uma ou mais bebidas produzidas por destilação de grãos, frutas ou vegetais que já passaram pela fermentação alcoólica e que podem ser misturados com outros ingredientes, como frutas, xaropes ou cremes. Os *cocktails* variam amplamente entre as regiões do mundo, e muitos sites publicam receitas originais e as suas próprias interpretações de *cocktails* mais antigos e famosos. A partir de meados da década de 2000, surgiu a cultura dos *cocktails* por meio do estilo que mistura *cocktails* tradicionais com outros ingredientes novos. No mundo moderno e na era da informação, as receitas de *cocktails* são amplamente compartilhadas online. Os *cocktails* e bares que os servem são frequentemente cobertos e avaliados em revistas e guias de turismo, nesse sentido o tema escolhido para este trabalho foi sobre *cocktails*.

Este tema foi escolhido por ser um tema que o grupo achou com potencial de ser explorado e uma vez que existe grande conhecimento sobre esta área seria interessante construir uma aplicação que permitisse explorar e apresentar esse conhecimento numa forma de fácil interpretação e interativa.

2.2 Trabalhos Relacionados

Um dos trabalhos baseados na construção desta aplicação foi *TheCocktailDB* [2] uma base de dados aberta e partilhada de bebidas e *cocktails* de todo o mundo. *TheCocktailDB* foi construído em 2015 para fornecer uma fonte de dados gratuita para bebidas online, na esperança de que os programadores criassem aplicações e projetos interessantes com esta plataforma. Muitos programadores, desde então, contribuíram para o projeto e criaram aplicações que utilizam estes dados. *TheCocktailDB* também é usado por universidades para tecnologia de design e programação da web, pois tem uma API fácil de usar e conteúdo interessante.

Capítulo 3

Metodologia, Tecnologias Utilizadas e Dataset

3.1 Metodologia

No sentido de alcançar os objetivos definidos que foi explorar conhecimento, necessitou-se de dados com conhecimento sobre *cocktails*, com auxílio de uma ontologia foi possível especificar formalmente esta área de conhecimento através da definição de conceitos que lhe são subjacentes e relações entre estes.

Deste modo, tendo a base do nosso conhecimento foi relevante entender brevemente a metodologia usada para a implementação da aplicação pretendida. Primeiro, foi necessário identificar quais os dados a utilizar e efetuar um tratamento sobre estes, de forma a poder construir uma ontologia o mais clara e simples possível. Posteriormente, de forma a ser possível a consulta deste conhecimento foi necessário integrá-lo numa base de dados específica, e por último o desenvolvimento da aplicação que permitiu a apresentação da informação.

3.2 Tecnologias Utilizadas

No contexto referido, criou-se uma ontologia com a ferramenta *Protégé* que fornece uma interface gráfica para definir ontologias. Deste modo, a informação é inserida numa base de dados orientada a grafos, *Ontotext GraphDB*, que permite a descoberta do conhecimento, sendo compatível com *RDF* e *SPARQL* e ainda se encontra disponível como um *cluster* de alta disponibilidade. A escolha recaiu neste paradigma, uma vez que, as ontologias são consideradas o input ideal para este tipo de base de dados.

De forma a que aplicação permita níveis de acesso, juntamente com um registo das ações dos utilizadores foi necessário garantir a persistência dos dados, para isso criou-se um documento com as respetivas coleções numa base de dados orientada a documentos, *MongoDB*, para a concretização desse objetivo realizou-se uma nova conexão do serviço de dados com outro tipo de base de dados.

Assim, a aplicação foi construída utilizando o conceito denominado *microservices* que segue uma abordagem arquitetónica e organizacional, esta é composta por pequenos serviços independentes que se comunicam usando *API's* (*Application Programming Interface*) bem definidas. Esta abordagem facilitou a escalabilidade e agilizou o desenvolvimento da aplicação, habilitando a inovação e acelerando o tempo de execução do projeto.

Neste sentido foi construído um serviço de dados baseado na *framework Open Source, Express*, disponibilizada pelo *Node.js* que recolhe a informação do *Ontotext GraphDB* usando *queries SPARQL* para o outro serviço criado que foi uma interface que controla o processo de interação com o utilizador e apresenta a informação numa interface web interativa, baseada na *framework Open Source, Express*, auxiliada com sintaxe *Pug* disponibilizada pelo *Node.js*.

3.3 Dataset

Os dados utilizados para este projeto foram obtidos de duas fontes. O primeiro no formato *Comma-separated values* (CSV) denominado *Cocktails (Hotaling & Co.)* foi retirado da plataforma *Kaggle* [3] que contém *cocktails* retirados da empresa *Hotaling & Co* que é uma importadora e destiladora de licores de São Francisco, Estados Unidos da América. Este *dataset* contém mais de 600 *cocktails*, onde cada registo diz respeito a uma bebida, à proveniência desta (criador e local), ingredientes que a compõem, instruções e guarnições.

O outro conjunto de dados utilizado foi obtido da API disponibilizada pelo *TheCocktailDB* através de um *script* na linguagem de programação *Python* onde se acedeu recursivamente a uma rota que disponibiliza os *cocktails* por letra alfabética e construiu-se um *dataset* também em CSV com a informação mais relevante (categoria, preparação, ingredientes, etc).

Finalmente, para obter o conjunto de dados base deste projeto recorreu-se novamente a outro *script* na linguagem de programação *Python* onde se juntou os dois *datasets* num CSV resultando assim num total de mais de 1000 *cocktails*. Cada registo contém assim informação sobre:

- *Cocktail Name*, nome da bebida;
- *Category*, categoria da bebida;
- *Alcoholic*, se a bebida possui álcool ou não;
- *Glassware*, o tipo de copo recomendável para servir a bebida;
- *Preparation*, instruções de preparação em inglês;
- *PreparationDE*, instruções de preparação em alemão;
- *PreparationIT*, instruções de preparação em italiano;
- *DrinkThumb*, URL para a fotografia da bebida;
- *Ingredients*, lista de ingredientes separados por ',' juntamente com a quantidade e medida;
- *Bartender*, criador da bebida;
- *Bar/Company*, bar ou empresa associada à bebida;
- *Location*, local associado à bebida;
- *Garnish*, lista de ingredientes separados por ',' que enfeitam a bebida;
- *Notes*, notas que descrevem o cocktail;

Capítulo 4

Implementação

4.1 Ontologia

Uma ontologia é um modelo de dados que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre estes, sendo que a mesma pode ser vista como uma especificação formal de conhecimento, i.e., capaz de ser compreendida por humanos e máquinas. Uma ontologia geralmente é descrita como o conjunto de:

- **Classes** : Normalmente organizadas em taxonomias, as classes representam algum tipo de interação da ontologia com um determinado domínio. Podem conter indivíduos, outras classes, ou uma combinação de ambos.
- **Atributos** : Cada atributo é utilizado para armazenar informação que é específica para o objeto associado a ele;
- **Relações**: Representam o tipo de interação entre os elementos do domínio (classes) e os indivíduos;
- **Indivíduos/Objetos**: São utilizados para representar elementos específicos, isto é, os próprios dados da ontologia.

O *Resource Description Framework* (RDF) é uma linguagem de propósito geral para caracterizar informações na web, representa as informações usando triplos semânticos, que compreendem um sujeito, um predicado e um objeto. Uma das linguagens para expressar os dados no modelo de dados RDF é a sintaxe *Turtle*, uma vez que é de fácil leitura e escrita. O grupo decidiu construir a ontologia nesta sintaxe uma vez que permite agrupar triplos e fornece maneiras de abreviar essas informações.

Com base na informação contida no dataset e uma breve análise da mesma, foi desenvolvida uma ontologia preparada para conter a informação presente no dataset. Para isto utilizou-se a ferramenta *Protégé*, definindo a seguinte ontologia:

- **Classes**
 - Bar_Company, bar ou empresa à qual a bebida está associada;
 - Bartender, quem cria a bebida;
 - Category, representa a categoria da bebida;
 - Cocktail, representa a bebida;
 - Glassware, representa o tipo de copo que serve a bebida;
 - Ingredient, representa um ingrediente utilizado numa bebida;
 - Location, representa a localização onde a bebida foi criada;
 - QuantityOfIngredient, representa a quantidade de ingrediente a utilizar numa bebida;

- **Relações (Object Properties)**

- asGarnishCocktail, ingrediente que enfeita uma bebida;
- associatedWithBar, bebida está associada a um bar ou empresa;
- createCocktail, *bartender* que cria uma bebida;
- createdByBartender, bebida criada por um *bartender*, relação inversa de *createCocktail*;
- createdInLocation, bebida criada numa localização;
- garnishWithIngredient, bebida guarnificada por um ingrediente, relação inversa de *asGarnishCocktail*;
- hasCategory, bebida possui uma categoria;
- haveCocktail, categoria possui bebidas, relação inversa de *hasCategory*;
- haveQuantity, ingrediente possui quantidades;
- isAssociatedToCocktail, bar ou empresa que está associado a bebidas, relação inversa de *associatedWithBar*;
- isUsedCocktail, quantidade de ingrediente utilizada numa bebida;
- needQuantity, bebida necessita de uma determinada quantidade de ingrediente, relação inversa de *isUsedCocktail*;
- originOfCocktail, uma localização é origem de várias bebidas, relação inversa de *createdInLocation*;
- serveCocktail, o tipo de copo que serve várias bebidas;
- serveInGlassware, bebida servida num tipo de copo, relação inversa de *serveCocktail*;
- useIngredient, quantidade relativa a um ingrediente, relação inversa de *haveQuantity*.

- **Atributos (Data Properties)**

- alcoholic, se a bebida é alcoólica ou não;
- drinkName, o nome da bebida;
- drinkThumb, o link da imagem da bebida;
- measure, unidade de medida associada a uma determinada quantidade de ingrediente;
- notes, notas de texto sobre cada bebida;
- preparationDE, preparação da bebida em alemão;
- preparationEN, preparação da bebida em inglês;
- preparationIT, preparação da bebida em italiano;
- quantity, quantidade de ingrediente;
- strBar, nome do bar ou empresa;
- strBartender, nome do *bartender*;
- strCategory, nome da categoria;
- strGlass, nome do tipo de copo;
- strIngredient, nome do ingrediente;
- strLocation, nome da localização;

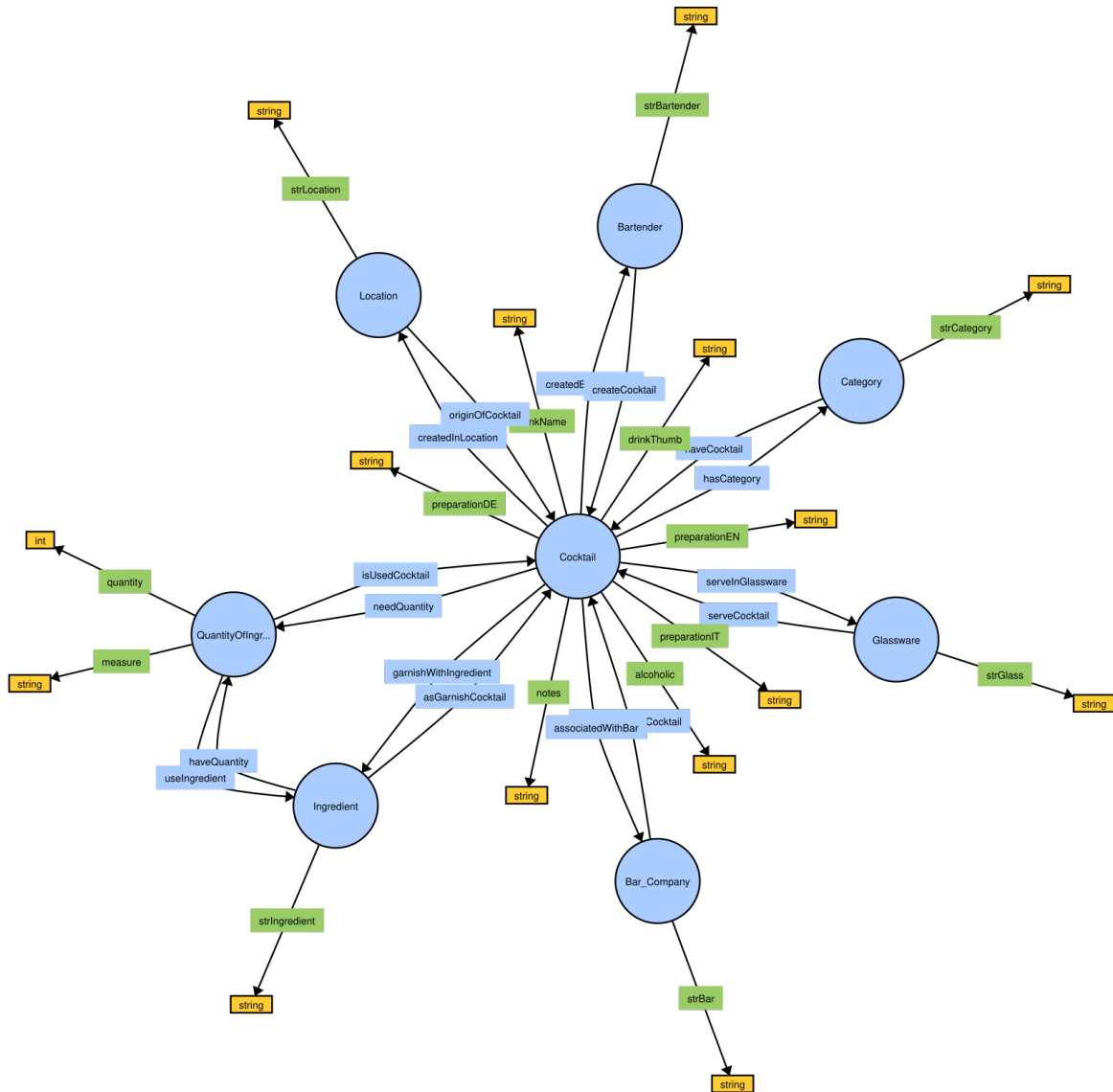


Figura 4.1: Estrutura da ontologia, imagem obtida através de *WebVOWL*.

4.2 Tratamento dos Dados

Após ter a ontologia criada, foi possível passar à fase seguinte do tratamento dos dados. Para tratar os dados e os juntar à ontologia criada desenvolveu-se um *script* em *Python* para tratar a informação.

Nesse sentido percorreu-se o *dataset* linha a linha, retirando a informação das colunas pretendidas e com recurso às estruturas disponibilizadas pela linguagem utilizada, mais propriamente os dicionários foi possível gerar os *individuals* na sintaxe *Turtle* com a informação recolhida. Depois de gerar todos os *individuals*, concatenou-se o resultado com o ficheiro da ontologia apresentada previamente.

Posteriormente, de forma a tornar a ontologia o mais completa possível recorreu-se novamente ao *Protégé* onde se inferiu novas relações através do *Reasoner* fornecido por este software, assim foi possível completar ainda mais a ontologia criada permitindo utilizar todas as relações previamente referidas. Deste modo, importou-se o ficheiro inferido para um novo repositório no *GraphDB*.

4.3 Aplicação

1. Serviço de Dados

Para este serviço criou-se um servidor *NodeJS* que recebe os pedidos efetuados pela interface aos quais vai devolver a resposta retirada do *GraphDB*.

Este serviço é composto por uma componente de consulta das informações presente na base de dados usando *queries SPARQL*, capazes de satisfazer os pedidos necessários da interface. Desta forma, o serviço de dados efetua pedidos à API disponibilizada pelo *GraphDB* que retorna os resultados das *queries* efetuadas em *JSON*, mas antes de devolver a resposta efetua um pré-processamento à informação recebida de forma a tornar mais clara e útil para a interface.

Foram desenvolvidas consultas sobre todas as classes apresentadas previamente, excepto a classe *QuantityOfIngredient*. Cada classe possui uma consulta que permite obter todos os elementos dessa classe juntamente com a quantidade de cocktails que estão associados, bem como, quantos *cocktails* não possuem nenhuma relação com essa classe. A outra consulta relativa a cada classe permite obter informação sobre um determinado indivíduo dessa classe, caso seja um *Cocktail* é apresentado todos os detalhes desse elemento, contudo se for um elemento de outra classe são apresentados alguns detalhes das bebidas que estão associadas a este elemento. De seguida apresentamos um exemplo referente à consulta das diferentes categorias, as restantes classes são idênticas, no caso da classe *Cocktail* é consultado mais informação:

```
SELECT ?idcategory ?strCategory (COUNT(?idcocktail) AS ?total) WHERE {
    ?idcocktail a :Cocktail .
    ?idcocktail :hasCategory ?idcategory .
    ?idcategory :strCategory ?strCategory .
} GROUP BY ?idcategory ?strCategory ORDER BY DESC (?total) `;

SELECT (COUNT(?idcocktail) AS ?total) WHERE {
    ?idcocktail a :Cocktail .
MINUS { ?idcocktail :hasCategory ?idcategory .}
}
```

De seguida apresentamos um exemplo referente à consulta de uma categoria, as restantes classes são idênticas, com excepção da classe *Cocktail*:

```
SELECT ?drinkid ?drinkname ?strGlass (COUNT(?idIngredient) AS ?totIngredients) WHERE {
    ${req.params.id} a :Category .
    ${req.params.id} :haveCocktail ?drinkid .
    ?drinkid a :Cocktail .
    ?drinkid :drinkName ?drinkname .
    OPTIONAL { ?drinkid :serveInGlassware ?idglassware . ?idglassware :strGlass ?strGlass . }
    OPTIONAL { ?drinkid :needQuantity ?idQuantity . ?idQuantity :useIngredient ?idIngredient . }
} GROUP BY ?drinkid ?drinkname ?strGlass ORDER BY ?drinkid
```

Além de extrair e apresentar conhecimento o grupo também achou pertinente adicionar mais conhecimento, bem como, a sua eliminação. Nesse sentido o grupo desenvolveu uma outra componente de autorização que fornece à interface *tokens* de acesso permitindo a gestão da plataforma, assim foram desenvolvidas rotas que permitem adicionar novos elementos de cada classe à base de dados como também rotas de eliminação (com excepção da classe *QuantityOfIngredient* e *Ingredient*) ou atualização, nesta última operação apenas se permitiu a edição dos atributos da classe *Cocktail*. Para demonstração apresentamos um exemplo de inserção de um novo *bartender* ao sistema, juntamente com associação a um *cocktail*:

```
var id = req.body.name.replace(/[^A-Za-z0-9-]/+, '').replace('-', '_').toLowerCase()
INSERT DATA {
    ${id} rdf:type owl:NamedIndividual .
    ${id} a :Bartender .
    ${id} :strBartender "${req.body.name}" .
    ${req.body.cocktails} :createdByBartender ${id} .
    ${id} :createCocktail ${req.body.cocktails} .
}
```

De seguida apresentamos um exemplo referente à eliminação de um *bartender*:

```
DELETE {
  :${req.params.id} rdf:type owl:NamedIndividual .
  :${req.params.id} a :Bartender .
  :${req.params.id} :strBartender ?strBartender .
  ?idCocktail :createdByBartender :${req.params.id} .
  :${req.params.id} :createCocktail ?idCocktail .
} WHERE {
  :${req.params.id} rdf:type owl:NamedIndividual .
  :${req.params.id} a :Bartender .
  :${req.params.id} :strBartender ?strBartender .
  OPTIONAL { ?idCocktail :createdByBartender :${req.params.id} . }
  OPTIONAL { :${req.params.id} :createCocktail ?idCocktail . }
}
```

Por fim, para atualizar a informação é necessário eliminar e posteriormente inserir com a nova informação, para exemplificação apresentamos a *query* de atualização do link para a fotografia de uma bebida:

```
DELETE {
  :${req.body.idDrink} :drinkThumb ?drinkThumb .
} WHERE {
  :${req.body.idDrink} rdf:type owl:NamedIndividual .
  :${req.body.idDrink} a :Cocktail .
  OPTIONAL { :${req.body.idDrink} :drinkThumb ?drinkThumb . }
}

INSERT DATA { :${req.body.idDrink} :drinkThumb "${req.body.newdrinkThumb}" . }
```

Deste modo qualquer operação requerida pela interface de administração de uma determinada entidade segue o exemplo de uma das três *queries* acima apresentadas.

Tendo em conta o tema do projeto considerou-se que seria interessante a expansão desta aplicação para algo mais completo e robusto, isto é, em vez de só existir um administrador existirem outros níveis de acesso, onde podemos efetuar comentários e/ou sugestões a *cocktails* e *bartenders* e, ainda ter uma lista *cocktails* e *bartenders* favoritos, bem como, seleccionar aqueles bares ou locais visitados. Assim, com auxílio de um outro *script Python* criaram-se ficheiros JSON com os identificadores de cada uma das classes pretendidas *Cocktail*, *Bartender*, *Bar_Company* e *Location*, e com auxílio da ferramenta *mongoimport* povoou-se as respetivas coleções. Nesse sentido definiu-se no Serviço de Dados os modelos pretendidos, é de realçar que o modelo da coleção de *Cocktail* e *Bartender* são idênticos porque pretendem disponibilizar o mesmo tipo de funcionalidades, tais como, a coleção *Bar_Company* e *Location* são parecidos, com o seguinte formato:

```
{
  "_id": "idBartender / idCocktail",
  "name": "strBartender / strCocktail",
  "reviews": [
    {
      publication_date: "" ,
      author: "_id do user" ,
      review: "",
      classification: }
  ],
  "likes": {
    "count": 0, "authors": []
  },
  "dislikes": {
    "count": 0, "authors": []
  }
}
```

```
{
  "_id": "idBar_Company / idLocation",
  "name": "strBar_Company / strLocation",
  "visits": { "count": 0, "authors": [] }
}
```

Para os níveis de acesso criou-se outra coleção com a informação básica de autenticação (email, nome, password e nível de acesso), desta forma cada utilizador pode-se autenticar ou registar, sendo validado pelo Serviço de Dados. Após identificado o utilizador é-lhe atribuído um *token* que permite efetuar as operações sobre as classes acima descritas, onde é registado na coleção devida a sua identificação.

2. Interface

Após ter a informação tratada e carregada no *GraphDB* juntamente com o Serviço de Dados que permite aceder à informação presente, a próxima e última etapa do trabalho passou pela criação de uma interface web interativa que disponibiliza informação de forma fácil e acessível, no qual foram usadas as *frameworks Express* e *Pug* disponibilizadas pelo *Node.js*.

Nesta interface, o utilizador ao aceder à página inicial, figura 4.2, é apresentada uma descrição da aplicação, também é possível encontrar uma barra lateral que possui o conjunto de operações permitidas nesta aplicação. Cada opção é um *link* para a página que permite efetuar a operação desejada, bem como, é sempre permitido efetuar uma pesquisa sobre um determinada bebida ou sobre um determinado *bartender*. O resultado desta pesquisa, presente na figura 4.3, devolve a lista de bebidas e *bartenders* cujos nomes correspondem a parte ou totalidade do que foi introduzido pelo utilizador.

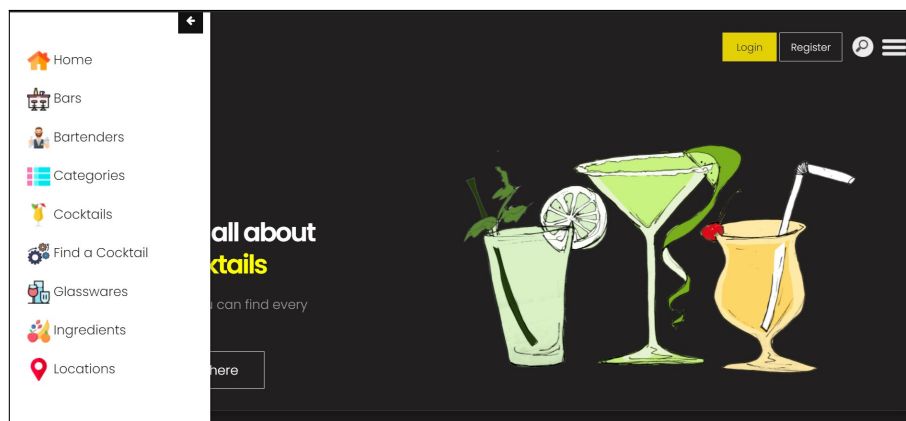


Figura 4.2: Página inicial da aplicação.

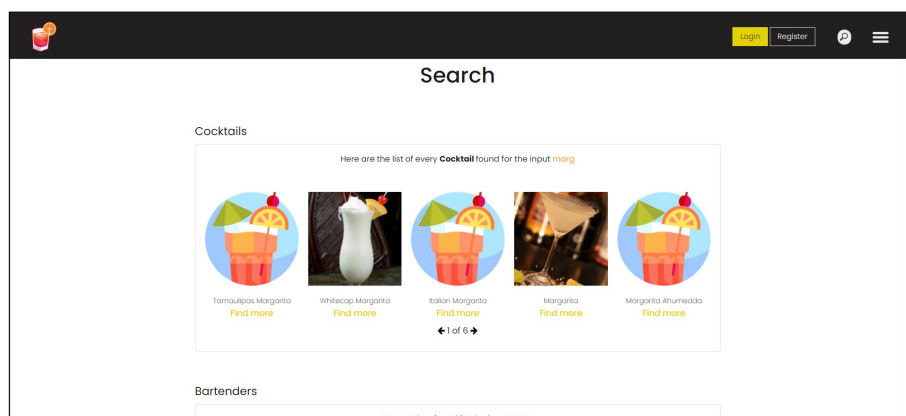


Figura 4.3: Resultado de uma pesquisa sobre um determinada bebida e sobre um determinado *bartender*.

Na barra lateral é possível aceder à lista de indivíduos das classes presentes na ontologia criada, com exceção da classe *QuantityOfIngredient*. Em cada uma destas listas, com exceção da lista dos *Cocktails*, é possível observar uma tabela (figura 4.4) com paginação sobre os elementos dessa classe e o número de bebidas que estão associadas, sendo possível ordenar ou pesquisar. Cada elemento destas tabelas remete à página individual, figura 4.5, onde é possível observar com mais detalhe as bebidas associadas a este indivíduo, podendo obter mais detalhes de uma determinada bebida.

Type of Glassware	Number of Cocktails
Cocktail Glass	126
Highball Glass	116
Coupe	106
Old Fashioned Glass	85
Collins glass	61
Collins	56

Figura 4.4: Lista de indivíduos de uma classe presente na ontologia.

Cocktail
9 1/2 Weeks
A. J.
AI
Adam

Figura 4.5: Indivíduo de uma classe presente na ontologia.

Em relação à lista de *Cocktails* é apresentado uma tabela com paginação na qual é possível ordenar e pesquisar sobre esta. Em cada linha desta tabela, figura 4.6, podemos encontrar o nome da bebida, categoria, tipo de copo, o nome do *bartender* associado e se possui álcool ou não, clicando sobre o nome desta, é possível observar toda a informação relacionada com o cocktail desde os ingredientes até à sua preparação nas diferentes línguas cobertas neste projeto, nos elementos que sejam classes da ontologia é possível aceder à página individual de cada elemento.

Cocktail	Category	Type of Glassware	Bartender	Alcoholic
#106 Cocktail	No Category	Coupe	Chelsea Little	✗
10 to 7	No Category	Martini Glass	Gabriele Modica	✗
110 in the shade	Beer Category	Beer Glass	No Bartender	✓
151 Florida Bushwacker	Milk / Float / Shake	Beer mug	No Bartender	✓
155 Belmont	Cocktail	White wine glass	No Bartender	✓
1880's Americana	No Category	Old Fashioned Glass	No Bartender	✗

Figura 4.6: Lista de *Cocktails*.

Figura 4.7: Página individual de um *Cocktail*.

Uma outra funcionalidade desenvolvida consiste em selecionar até 3 ingredientes e encontrar as bebidas que satisfazem essa combinação, esta funcionalidade, presente na figura 4.8, é útil para qualquer utilizador pois permite com base nos ingredientes que tenha em casa encontrar uma bebida que utilize os mesmos.

Figura 4.8: Combinar ingredientes.

Em relação à parte de gestão de conhecimento foi realizado uma parte de administração onde nesta, numa primeira fase necessita de uma chave de acesso que é validada pelo Serviço de Dados e caso seja correta, é atribuído um *token* à interface baseado numa chave privada e pública permitindo autenticar assim a interface. Posteriormente é pedido ao administrador para introduzir a palavra-passe que permite a sua identificação, caso a introduza corretamente é gerado um novo *token* com um segredo no qual é guardado nas *cookies* o nível de acesso deste utilizador.

Foram desenvolvidas operações de manipulação das classes, para isso, é apresentada uma página com a lista de elementos das classes onde em cada uma é possível aceder à tabela de cada classe, sendo possível assim a eliminação, como também, à adição de novos indivíduos a estas classe, é de realçar que a informação a inserir nestes novos elementos está limitada pelos elementos das outras classes. Em cada elemento destas listas é possível aceder a mais informação sobre um elemento, onde no caso dos *Cocktails* é permitido editar informação, atualizar a foto da bebida, acrescentar e/ou eliminar ingredientes que tanto enfeitam ou fazem parte da bebida. Nos restantes casos apenas é possível associar ou desassociar bebidas, com exceção da classe *Ingredient* onde apenas se pode criar novos elementos desta classe.

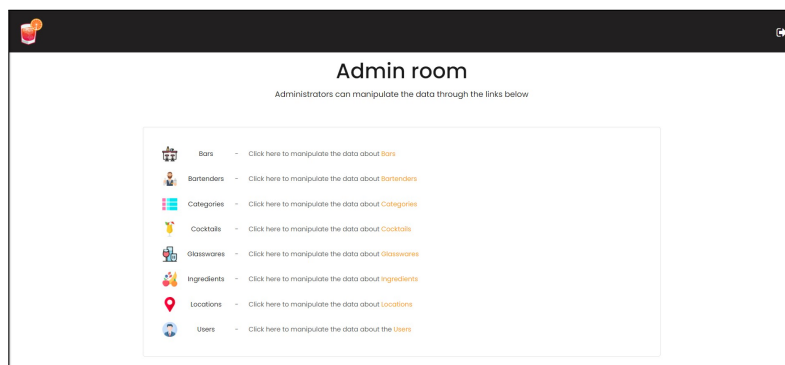


Figura 4.9: Gestão de conhecimento.

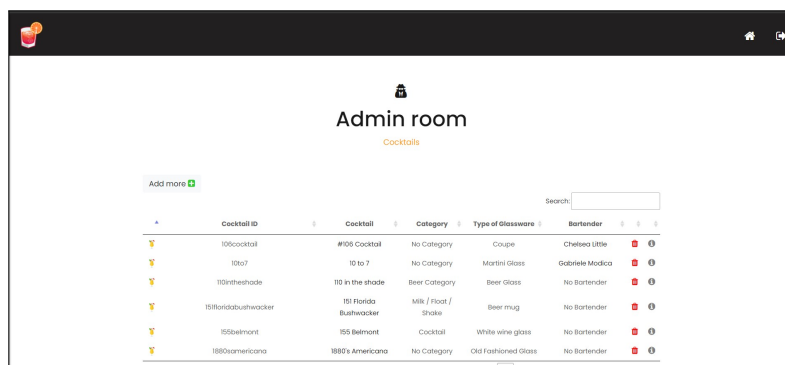


Figura 4.10: Gestão dos elementos da classe *Cocktail*.

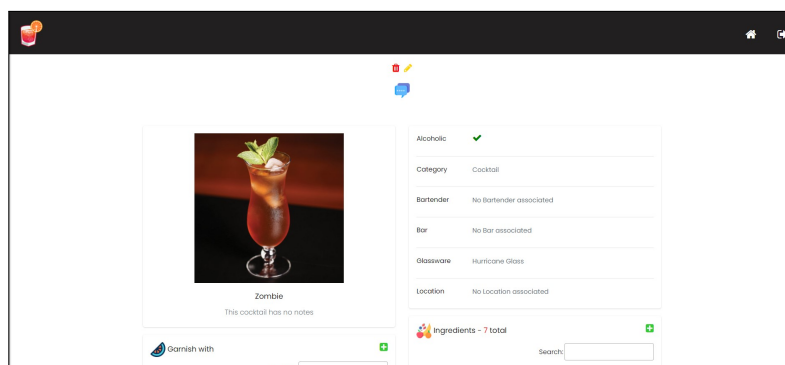


Figura 4.11: Gestão de um elemento da classe *Cocktail*.

No sentido de o utilizador participar na aplicação, isto é, comentar *Cocktails* e *Bartenders*, gostar ou não gostar de elementos dessas classes deve estar previamente autenticado, tal como, para marcar locais ou bares já visitados. Tendo sempre a possibilidade de aceder ao seu perfil onde são apresentados os elementos de que gostou ou visitou.

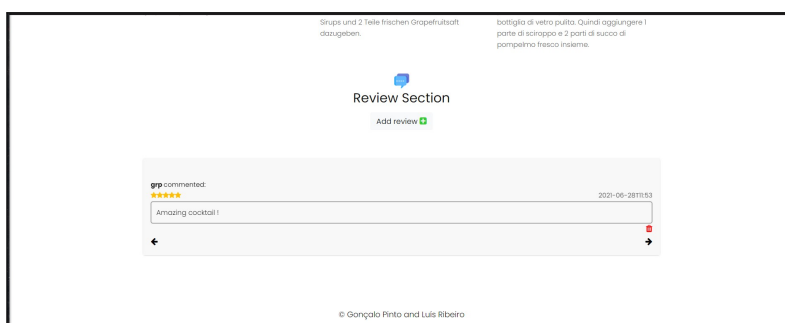


Figura 4.12: Comentários introduzidos pelos utilizadores.

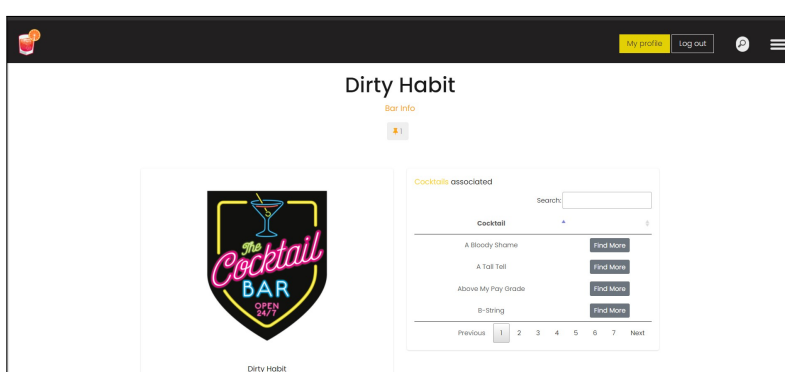


Figura 4.13: Locais visitados pelos utilizadores.

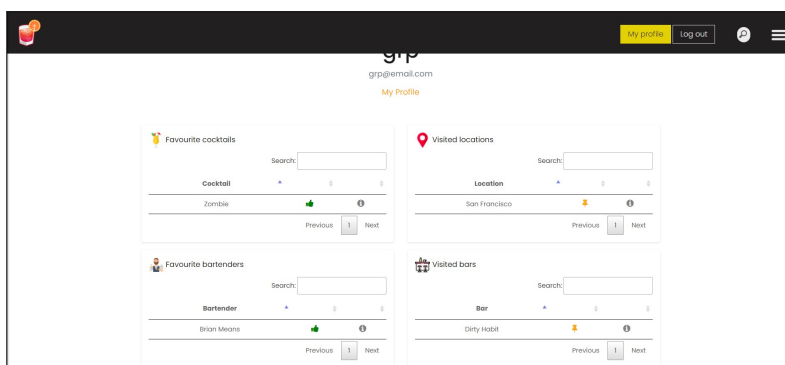


Figura 4.14: Perfil de um utilizador.

Capítulo 5

Conclusão e Trabalho Futuro

O presente relatório descreveu, de forma sucinta, o trabalho de especificação de conhecimento de forma a conseguir construir uma aplicação informática que tire partido desse conhecimento.

Durante a elaboração deste trabalho surgiram algumas dificuldades que, com o esforço e entusiasmo do grupo pelo produto final, acabaram por ser ultrapassadas. Entre as quais destacam-se o desafio que foi lidar com a inconsistência dos dados utilizados, especialmente extrair a quantidade e medida de ingrediente que uma determinada bebida necessita.

Numa perspetiva futura, considera-se que seria interessante a expansão desta ontologia para algo mais completo e robusto, isto é, aumentar a quantidade de *cocktails* com as respetivas relações e atributos.

Após a realização deste trabalho, o grupo ficou consciente da importância da representação do conhecimento através de ontologias. Consideramos que os objetivos propostos com a realização deste trabalho foram cumpridos, bem como, a consolidação dos conhecimentos em processamento e representação de conhecimento.

Por fim, o grupo espera que os conhecimentos obtidos e consolidados sejam de enorme utilidade tendo em conta uma perspetiva futura.

Bibliografia

- [1] *A brief history of cocktails*. 2020. URL: <https://www.diffordsguide.com/encyclopedia/2294/cocktails/a-brief-history-of-cocktails> (acedido em 25/06/2021) (ver p. 4).
- [2] *TheCocktailDB*. 2021. URL: <https://www.thecocktaildb.com/> (acedido em 25/06/2021) (ver p. 4).
- [3] *Cocktails (Hotaling & Co.)*. 2019. URL: <https://www.kaggle.com/shuyangli94/cocktails-hotaling-co> (acedido em 25/06/2021) (ver p. 6).