



UNIVERSIDADE DO MINHO

MESTRADO EM ENGENHARIA INFORMÁTICA

GRAMÁTICAS NA COMPREENSÃO DE SOFTWARE

---

# Sistema de Perguntas e Respostas

Trabalho prático

---

**Autores:**

Frederico Pinto - 73639

Rui Vieira - 74658

14 de Janeiro de 2019

# *Conteúdo*

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Problemetização e Implementação</b>	<b>3</b>
2.1	Descrição Informal do Problema . . . . .	3
2.2	DSL . . . . .	3
2.3	Concepção da Resolução . . . . .	4
2.4	Codificação e Testes . . . . .	4
<b>3</b>	<b>Conclusão</b>	<b>6</b>

# 1. *Introdução*

Na unidade curricular de Gramáticas na Compreensão de Software, foi-nos proposto o desenvolvimento de um sistema de Perguntas e Respostas de um tema à nossa escolha. Após um período de reflexão sobre o assunto, optamos por escolher um tema sobre restauração, ou seja, iremos colocar na base de conhecimento informação sobre restaurantes desde pratos, bebidas, contactos, entre outros. Este tema pareceu-nos bastante pertinente e vantajoso para o utilizador.

Conforme a pergunta colocada, o sistema irá procurar à base de conhecimento informação sobre a pergunta feita e responderá sobre aquilo que sabe sobre a questão efetuada.

## 2. *Problemetização e Implementação*

### 2.1 Descrição Informal do Problema

A informação que vai ser recebida para guardar na base de conhecimento pode ser representada na forma de triplos, nos quais temos um restaurante que tem uma relação com determinado objeto. Para além disso, após esses triplos temos as perguntas em si que têm que ser lidas, analisadas e efetuadas à base de conhecimento.

### 2.2 DSL

Para captar a informação fizemos uma gramática capaz de analisar tanto a informação, para inserir na base de conhecimento, como as perguntas para efetuar no final. Apresentamos de seguida, o resultado da gramática efetuada.

Listing 2.1: DSL utilizada

---

```
1 grammar DSL;
2
3 sistema: (restaurante)+ perguntas
4         ;
5
6 restaurante: nome informacao
7             ;
8
9 nome : TEXTO
10      ;
11
12 informacao : triplo (',' triplo)*
13            ;
14
15 triplo: '{ 'tem' TEXTO '}'
16       | '{ 'contacto' TEXTO '}'
17       | '{ 'morada' TEXTO '}'
18       | '{ 'horario' TEXTO '}'
19       ;
20
21 perguntas: pergunta (',' pergunta)*
22          ;
23
24 pergunta : '[' TEXTO ']'
25          ;
```

---

Apresentamos agora uma linguagem válida para a gramática representada acima. Aparecendo primeiro o nome do restaurante e logo de seguida os triplos que são a informação para ser guardada, depois dos triplos aparecem as perguntas.

Listing 2.2: Linguagem resultante da DSL

---

```
1 Tasquinha{ tem "Cordon Blue"},
2 { tem "Arroz de feijao com Ovos" },
3 { tem "Bacalhau Frito"},
4 { tem "Frango de Churrasco"},
5 { tem "Coca Cola"},
6 { tem "Sprite"},
7 { tem "Agua"},
8 { tem "Ice Tea"},
9 { tem "Blimonada de Morango"},
10 { tem "Vinho Tinto"},
11 { tem "Vinho Branco"},
12 { tem "Cerveja"},
13 { morada "Rua do Monopolio n6"},
14 { contacto 253761123}
15
16 [ "A Tasquinha tem Coca Cola?"],
17 [ "Qual o horario da Tasquinha?"],
18 [ "O Sabores da Fruta tem mesmo fruta?"],
19 [ "Qual o contacto da Tasquinha?"],
20 [ "A Tasquinha encontra-se ao lado do Speedy?"]
```

---

## 2.3 Conceção da Resolução

Para construirmos a base de conhecimentos utilizamos várias estruturas de dados da linguagem *Java* como *HashMaps* e *ArrayLists*. A estrutura principal é um *HashMap(String, HashMap())*, em que a chave é o nome do restaurante e o valor é outro *HashMap*. Este último é um *HashMap(String, ArrayList)*, em que a chave é a acção entre o restaurante e o objeto, já o valor é um *ArrayList* que possui todos os pares (Objeto, Resposta) para a relação.

Utilizamos a API *Substring* do *Java* para facilitar a procura das palavras-chave para assim percorrermos a base de conhecimento para poder devolver uma resposta. Assim, procuramos na pergunta o nome do restaurante, depois uma acção e por fim um objeto.

## 2.4 Codificação e Testes

Após a criação da gramática de atributos com a implementação do código em que utilizamos atributos herdados e sintetizados, chegamos a um ponto final no desenvolvimento de software e podemos partir para uma fase de análises de testes, utilizamos diversos testes mas apresentamos neste relatório de desenvolvimento as seguintes questões.

Listing 2.3: Questões a efetuar

---

```
1 [ "A Tasquinha tem Rojoes?"],
2 [ "O Speedy tem algum tipo de vinho tinto?"],
3 [ "Qual a morada do Speedy?"],
4 [ "Qual o horario da Tasquinha?"]
```

---

```
5 [ "O Sabores da Fruta tem mesmo fruta?" ],  
6 [ "Qual o contacto da Tasquinha?" ],  
7 [ "A Tasquinha e ao lado do Speedy?" ]
```

---

```
"A Tasquinha tem Rojões?"  
Isto é o que sabemos sobre o assunto tem sobre Tasquinha:  
-> Cordon Blue;  
-> Arroz de feijão com Ovos;  
-> Bacalhau Frito;  
-> Frango de Churrasco;  
-> Coca Cola;  
-> Sprite;  
-> Água;  
-> Ice Tea;  
-> Blimonada de Morango;  
-> Vinho Tinto;  
-> Vinho Branco;  
-> Cerveja;  
  
"O Speedy tem algum tipo de vinho tinto?"  
Sim, é verdade!  
  
"Qual a morada do Speedy?"  
Isto é o que sabemos sobre o assunto morada sobre Speedy:  
-> Rua de Barros nº129;  
  
"Qual o horario da Tasquinha?"  
Isto é o que sabemos sobre o assunto horario sobre Tasquinha:  
  
"O Sabores da Fruta tem mesmo fruta?"  
Restaurante não existe! Reformule a questão!  
  
"Qual o contacto da Tasquinha?"  
Isto é o que sabemos sobre o assunto contacto sobre Tasquinha:  
-> 253761123;  
  
"A Tasquinha é ao lado do Speedy?"  
Não percebemos o que quer saber sobre o restaurante! Reformule a questão!
```

Figura 2.1: Respostas do sistema às perguntas

Como dissemos anteriormente, procurar as palavras-chave na pergunta segue uma ordem, nome do restaurante, acção e objeto. Caso se encontre as três palavras chave e a pergunta seja verdade, o sistema apenas devolverá a frase *"Sim, é verdade!"*, caso não reconheça o objeto irá mostrar tudo o que sabe sobre o restaurante e ação presente na pergunta. Caso não reconheça efetivamente nenhum restaurante presente na pergunta ou a ação entre restaurante/objeto, pede para reformular a pergunta, para assim tentar obter uma resposta mais correta.

### 3. *Conclusão*

Durante a realização do projeto utilizamos vários conteúdos lecionados durante o semestre, comprovando empiricamente a utilidade da utilização de gramáticas no desenvolvimento de software. Encontramos algumas dificuldades, como seria de esperar, pela falta de experiência no desenvolvimento de sistemas deste gênero. No entanto ficamos satisfeitos com trabalho desenvolvido e com confiança que no futuro estamos mais preparados não só para o desenvolvimento de sistemas Pergunta-e-Resposta mas também na utilização de gramáticas.