



Universidade do Minho
Escola de Engenharia

Web scraping + Ontologia

By Rafael Silva

Scripting no Processamento de Linguagem Natural

Universidade do Minho, Mestrado Integrado em Engenharia Informática,
4º Ano, 2º Semestre, Junho 2020



Fases do Projeto



WebSite - AutoEvolution

- Website contém a informação de todas as marcas a nível mundial.
- Possui todos os modelos de todas as marcas.
- Versões de cada um desses modelos.
- Especificações de cada versão.



ACURA

ACURA MDX A-Spec · ACURA ILX · ACURA TLX

7



IN PRODUCTION

43



DISCONTINUED



ALFA ROMEO

ALFA ROMEO Stelvio Quadrifoglio · ALFA ROMEO MiTo · ALFA ROMEO Giulia GTA

11



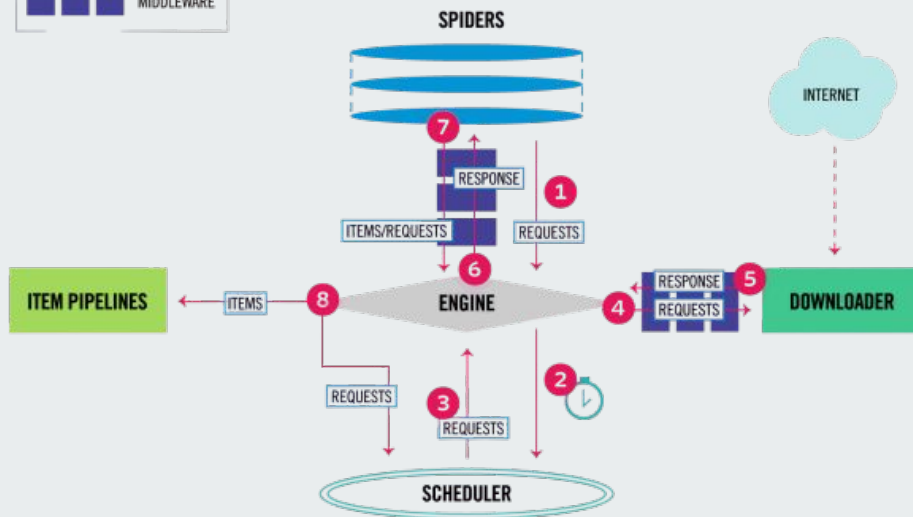
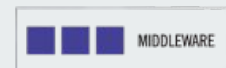
IN PRODUCTION

82



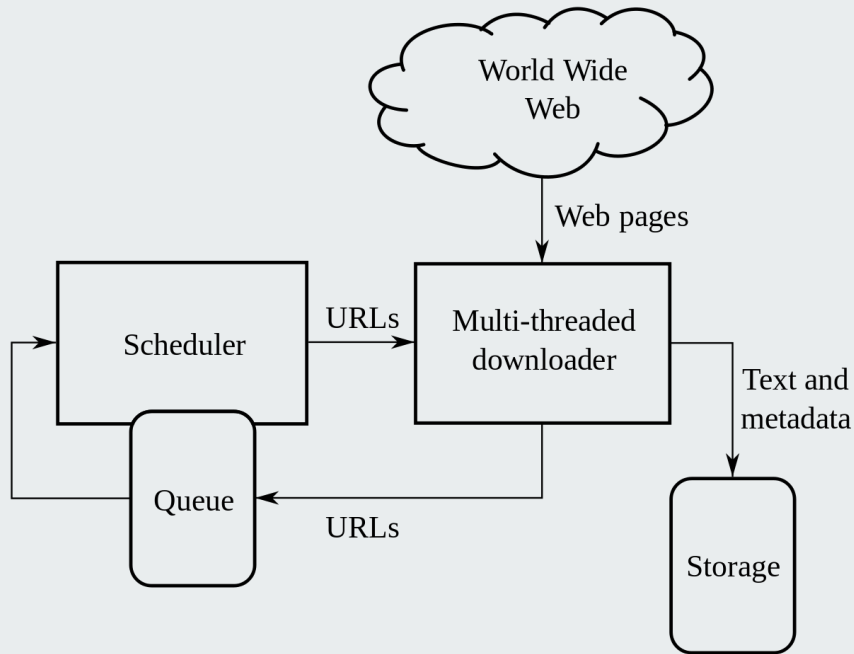
DISCONTINUED

Scrapy



- Consiste numa *framework* de *Web-Crawling*.
- A arquitectura do **scrapy** é construída em torno de "*spiders*", que são *crawlers* autônomos que recebem um conjunto de instruções.

Web-Crawler



- *Web-Crawler* é um programa que navega pela Internet de uma forma metódica e automatizada.
- O mesmo utiliza o código HTML da página fonte para fazer essa navegação.



Desenvolvimento - *Script*

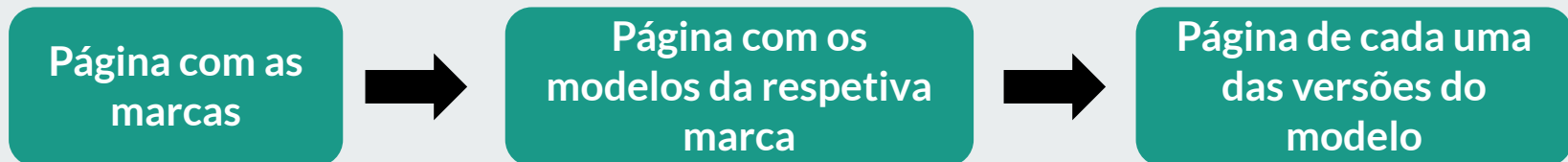
```
1 import scrapy
2 class AutoevolutionSpider(scrapy.Spider):
3     name = 'AutoEvolution'
4     allowed_domains = ['autoevolution.com']
5     start_urls = ['http://autoevolution.com/cars/']
6     def parse(self, response):
7         pass
```

- *Scrapy* configuração por default.
- array de *allowed_domains*
- array de *start_urls*



Navegação do *Web-Crawler*

- Primeiro o mesmo irá aceder ao *link* onde se encontram todas as marcas.
- De seguida irá recolher todas as informações.
- Visitar a *link da marca em questão* e recolher toda a informação necessária.
- Em seguida irá navegar para o *link* do correspondente a cada modelo e dentro deste irá também recolher dados.
- Por fim irá aceder ao *link* de cada versão do respetivo modelo.





Exemplo de um *parser*

```
1 ...
2 def parseBrand(self, response):
3     ...
4     CAR_LIST_SELECTOR = '.carmod'
5     for car in response.css(CAR_LIST_SELECTOR):
6         ...
7         URL_CAR = '.fl > a::attr(href)'
8         url_car = car.css(URL_CAR).extract_first()
9         model = {
10             ...
11             'url': url_car,
12             ...
13         }
14         if url_car:
15             request = scrapy.Request(url_car, callback=self.parseModelVersion, meta={
16                 'model': model, 'brand': brand})
17             yield request
18 ...
```



Tratamento de Dados

```
1 ...
2 class AutoevolutionScraperItem(scrapy.Item):
3     brand = scrapy.Field()
4     model = scrapy.Field()

1 import json
2 class AutoevolutionScraperPipeline:
3     def process_item(self, item, spider):
4         line = json.dumps(dict(item), indent=2) + ',\n'
5         self.file.write(line)
6         return item
7     def open_spider(self, spider):
8         self.file = open('autoevolution.txt', 'w')
9         line = '[\n'
10        self.file.write(line)
11    def close_spider(self, spider):
12        line = ']\n'
13        self.file.write(line)
14        self.file.close()
```

- Usar um objeto em *JSON*.
- Por cada *parser* guardam-se dados.
- Passar a informação de *parser* para *parser*.
- Completar o objeto final no último *parser*.
- Escrever esse objeto num ficheiro utilizando as *pipelines* do scrapy

Ontologia

The screenshot shows a web browser window with the URL <http://www.semanticweb.org/rafaelsilva/ontologies/2020/5/auto-evolution-complete>. The browser displays the 'Brand' ontology. The left sidebar shows the class hierarchy: **owl:Thing** (parent) has subclasses **Brand**, **Models**, and **Versions**. **Brand** has subclasses **Fuels** and **Classes**. The main content area shows the 'Brand' class with its description and instances. The instances listed are: ACURA, ACURA_CL, ACURA_CL_2_2L_5MT_145_HP_, ACURA_CL_3_2_Type_5, ACURA_ILX, ACURA_ILX_1_5L_Hybrid_CVT_111_HP_, ACURA_ILX_2_0L_5AT_150_HP_, and ACURA_ILX_2_4L_16V_i-VTEC_8AT_201_HP_.

- Possui quatro classes e duas subclasses.
- **Fuels**, **Brand** e **Classes** são as principais.
- **Models** e **Versions** são subclasses, a primeira de Brand e a segunda de Models.

Ontologia - Relações

Active ontology x Entities x Individuals by class x DL Query x

Datatypes Individuals

Data properties Annotation properties

Classes Object properties

Object property hierarchy ? ? ? ? ?

Asserted

owl:topObjectProperty

- classDoModelo
- combustivelUsado
- pertenceClass
- pertenceMarca
- pertenceModelo
- temModelo
- temVersao
- usaCombustivel

Annotations: temModelo

Annotations +

Ch: ? ? ? ? ? Description: temModelo ? ? ? ? ?

☐ Function: ☐ Inverse function: ☐ Transitive: ☐ Symmetric: ☐ Asymmetric

Inverse Of +

pertenceMarca ? @ x o

Domains (intersection) +

Brand ? @ x o

Ranges (intersection) +

Git: master To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

- Possui oito relações.
- Quatro dessas relações são o inverso da correspondente.
- **pertenceClass** - relaciona **Models** e **Classes**, o seu inverso é **classDoModelo**.
- **temModelo** - relaciona **Brands** e **Models**, o seu inverso é **pertenceMarca**.
- **temVersao** - relaciona **Models** e **Versions**, o seu inverso é **pertenceModelo**.
- **usaCombustivel** - relaciona **Models** e **Fuels** o seu inverso é **combustivelUsado**.



Povoamento - Ontologia (Exemplo)

```
1 def populate_model(data):
2     with open("auto_evolution.ttl", "a") as auto_evolution:
3         for model in data:
4             brand_name = model['nameBrand']
5             ...
6             model_aux = '''
7 ### http://www.semanticweb.org/rafaelsilva/ontologies/2020/auto-evolution#{0}
8 :{0} rdf:type owl:NamedIndividual ,
9       :Models ;
10      :pertenceClass :{1} ;
11      :pertenceMarca :{2} ;
12      :usaCombustivel {3}
13      :imgModel "{4}"^^xsd:string ;
14      :modelYears "{5}"^^xsd:string ;
15      :nameModel "{0}"^^xsd:string ;
16      :nrGenerationsModel "{6}"^^xsd:string ;
17      :urlModel "{7}"^^xsd:string .
18 ''' .format(model_name, model_class, brand_name, fuels_aux, model_img, modelYears,
19             model_nrGenerations, model_url)
20     ...
```



GraphDB

Class	Links	
:Models	60K	↔
:Brand	29K	↔
:Fuels	11K	↔
:Classes	8K	↔
:Versions	6K	↑



Universidade do Minho
Escola de Engenharia

Web-Scraping + Ontologia

By Rafael Silva

Scripting no Processamento de Linguagem Natural

Universidade do Minho, Mestrado Integrado em Engenharia Informática,
4º Ano, 2º Semestre, Junho 2020