



UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO DE ENGENHARIA INFORMÁTICA

Interoperabilidade Semântica

Trabalho Prático

Adriana Guedes	A74545
Bruno Sousa	A74330
Marco Barbosa	A75278
Ricardo Certo	A75315

18 de Junho de 2018

Conteúdo

1	Introdução	2
2	Caso de Estudo	3
3	Programas	4
3.1	Programa 1	4
3.1.1	Como usar	4
3.2	Programa 2	5
3.2.1	Como usar	5
3.3	Programa 3	5
3.3.1	Como usar	5
4	Interface Web	6
4.1	Interface programa 1	6
4.2	Interface programa 2 e 3	7
5	Performance	8
5.1	Performance Programa 1	8
5.2	Performance Programa 2	8
5.3	Performance Programa 3	8
5.4	Análise Comparativa de Performance	8
6	Conclusão e Trabalho Futuro	9

1 Introdução

No âmbito desta Unidade Curricular, foi-nos proposto o desenvolvimento de um trabalho prático que consiste, essencialmente, na elaboração de um sistema aonde serão aplicados todos os métodos de Interoperabilidade Semântica, em particular arquiteturas baseadas em serviços em Web. Para isso vamos implementar dois tipos de arquiteturas: uma que funciona em modo online e outra que funciona de modo offline.

O sistema desenvolvido, por isso, deverá ser capaz de, dado um ou vários orcid's válidos de ir buscar toda a informação sobre um ou mais investigadores e apresentá-la numa página web, analisando também a performance desse sistema.

2 Caso de Estudo

Para o trabalho realizado, a grande fonte de informação provém dos designados *orcids* (*Open Researcher and Contributor ID's*), que não são mais do que identificadores próprios atribuídos a académicos/investigadores, facilitando o registo e automatização de publicação de trabalhos, artigos, etc.

Sendo assim, pretendem-se representar através de uma página web os vários atributos que podem ser recolhidos por cada *orcid*: número *orcid*, *putCode*, *lastModifiedDate*, título, ano, *localpub* e *scopus*.

Para recolher tal informação, foram utilizadas as *API's* disponibilizadas pelo *orcid*, informação essa guardada numa base de dados relacional *MySql* e sendo todos os programas implementados segundo a linguagem de programação *python*.

3 Programas

3.1 Programa 1

O programa fica a correr em background e sempre que quisermos obter resultados sobre um orcid específico temos de o adicionar para o podermos analisar. Ao analisarmos um ou vários orcid's guardamos a sua respetiva informação numa base de dados, para depois representarmos a sua informação numa tabela numa *web page*. Optamos, por guardar a informação numa base de dados ao invés de guardar num JSON pelo facto de um artigo poder ter vários contribuidores e assim facilitar o relacionamento entre eles. Para a gerar a pagina web foi usado um script em python que acede á base de dados e cria a respectiva página html.

3.1.1 Como usar

De seguida vamos apresentar uma breve lista de comandos que ajudam no uso do programa 1.

Start background syncing process

```
python background
```

List ORCID being analyzed

```
python background -l
```

Add ORCID('s) to be analyzed

```
python background -a 0000-0003-4121-6169
python background -a 0000-0003-4121-6168 0000-0003-4121-6169
0000-0003-4121-6170
```

Remove ORCID('s) from being analyzed

```
python background -r 0000-0003-4121-6169
python background -r 0000-0003-4121-6168 0000-0003-4121-6169
0000-0003-4121-6170
```

3.2 Programa 2

Este programa é responsável por usar o *webservice* e representar logo a informação recolhida numa página *web*. Para isto foi usada a biblioteca ReactJS, que faz parse da informação obtida através do *webservice* e apresenta-a de imediato.

3.2.1 Como usar

Vamos apresentar a forma como usamos o segundo programa.

Iniciar página

```
npm start
```

3.3 Programa 3

Este programa foi desenvolvido após o grupo considerar que a implementação do primeiro programa não foi a melhor. Neste foi desenvolvido um servidor que trata de actualizar a base de dados e onde foi implementada uma API em rest. Foi também desenvolvido um front-end com o mesmo design que o segundo programa usando a biblioteca react que apenas faz pedidos ao servidor usando a API Rest.

3.3.1 Como usar

Vamos apresentar a forma como usamos o segundo programa.

Iniciar servidor

```
npm start
```

Iniciar página

```
npm start
```

4 Interface Web

A interface web vai ser aonde vamos visualizar as informações recolhidas pelos dois programas que mencionamos anteriormente. Essas informações vão estar predispostas em tabelas de uma forma user friendly com o objetivo de utilizador comum as conseguir interpretar e concluir algo sobre elas.

De seguida podemos ver algumas imagens da interface web implementada.

4.1 Interface programa 1



Figura 1: Interface prog.1

4.2 Interface programa 2 e 3

IS - Sistema ORCID					
orcid: 0000-0000-0000-0000	Inserir	Orcid	Titulo	Ano	Scopus
TODOS		0000-0003-4121-6169	Iron Value Classification in Patients Undergoing Continuous Ambulatory Peritoneal Dialysis using Data Mining	2018	
0000-0003-4121-6169	x	0000-0003-4121-6169	Comparison of two control programs of the VVER-1000 nuclear power unit using regression data mining models	2017	85043974669
0000-0003-0549-3097	x	0000-0003-4121-6169	New approach to an openEHR introduction in a portuguese healthcare facility	2018	85045320739
		0000-0003-4121-6169	Step towards a pervasive data system for intensive care medicine	2018	85045347385
		0000-0003-4121-6169	Mobile collaborative augmented reality and business intelligence: A system to support elderly people's self-care	2018	85045332393
		0000-0003-4121-6169	Step Towards Prediction of Perineal Tear	2017	85033453791
		0000-0003-4121-6169	Magni - A framework for developing context-aware mobile applications	2017	85018578020
		0000-0003-4121-6169	Predict sepsis level in intensive medicine - Data mining approach	2013	84876221428
		0000-0003-4121-6169	Step towards paper free hospital through electronic health record	2013	84876207927
		0000-0003-4121-6169	Knowledge Discovery from Surgical Waiting lists	2017	85040217774
		0000-0003-4121-6169	Predicting the need of Neonatal Resuscitation using Data Mining	2017	85033438494

Figura 2: Interface prog.2 e 3

5 Performance

A performance em interoperabilidade semântica é um aspeto importante a termos em conta, pois é ela que nos ajuda a perceber se um programa é ou não eficiente e se deve ser melhorado ou não. A métrica de performance que vamos usar vai ser o tempo de execução dos programas, desde que vai buscar as informações até que as apresenta na página web.

5.1 Performance Programa 1

A performance obtida para o primeiro program foi a seguinte: 0.3 segundos

5.2 Performance Programa 2

A performance obtida em média para o primeiro program foi a seguinte: 1.5 segundos

5.3 Performance Programa 3

A performance obtida em média para o primeiro program foi a seguinte: 0.6 segundos

5.4 Análise Comparativa de Performance

Podemos concluir, que o programa 1 apresenta uma maior performance relativamente ao programa 2. No entanto também podemos observar que apesar de um pouco mais lento, dada a arquitectura cliente-servidor, o programa 3 é ainda mais rápido que o programa 2. Com a vantagem em relação ao programa 1 de não ser necessário estar a correr o script de acesso á base de dados sempre que seja necessário actualizar a informação. A grande diferença de tempos deve-se ao download das informações da API.

6 Conclusão e Trabalho Futuro

Este trabalho prático foi bastante importante na medida em que permitiu aprofundar conhecimentos sobre a matéria abordada nas aulas, uma vez que para a implementação do sistema proposto foi necessário recorrer a diversos aspetos lecionados, desde o uso das API's dos ORCID's até às análises de performance de um programa. Também aprofundamos conhecimento no que toca a implementação de API's em Rest e pedidos de cliente-servidor através desta.

Sentimos maiores dificuldades na representação da informação nas *web pages*, devido a não possuímos um conhecimento muito vasto de *front-end*.

Como trabalho futuro, poderíamos acrescentar mais informações à *web page* do tipo fotografia dos contribuidores, uma pequena bibliografia sobre eles, apresentar outro tipo de informações/trabalhos que eles tenham realizado, também poderíamos fazer com a *web page* ficasse sempre ativa com um domínio próprio em que já possuiria informações adicionais mencionadas em cima.