 **INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

Projeto e Seminário

Licenciatura em Engenharia Informática e Computadores

Ana Carolina Baptista

[41487@alunos.isel.ipl.pt](mailto:41487@alunos.isel.ipl.pt)

960314580

Eliane Almeida [41467@alunos.isel.ipl.pt](mailto:41467@alunos.isel.ipl.pt)

960271968

Orientadores:

Cátia Vaz, ISEL, [cvaz@cc.isel.ipl.pt](mailto:cvaz@cc.isel.ipl.pt)

José Simão, ISEL, [jsimao@cc.isel.ipl.pt](mailto:jsimao@cc.isel.ipl.pt)

Alexandre P. Francisco, IST, [aplf@ist.utl.pt](mailto:aplf@ist.utl.pt)

19 de Março de 2018

# Introdução

Atualmente, com o grande crescimento e propagação de dados na internet, surge a necessidade de que a informação seja descrita e transmitida por meio de uma linguagem *standard*, sendo esta de fácil entendimento tanto para computadores quanto para humanos.

Uma das técnicas de descrição de informação que se está a tornar muito popular é baseada em ontologias [1]. Esta permite especificar explicitamente uma conceptualização ou um conjunto de termos de conhecimento para um domínio particular. Apesar da popularidade das ontologias, há em geral dificuldade em transformar o conhecimento pré-definido num caso concreto.

Na área da bioinformática, existem recursos científicos que necessitam de ser partilhados entre a comunidade científica por meio de ontologias. Sendo as ontologias normalmente definidas através de OWL [2] (*Web Ontology Language*), em várias situações poderá não ser uma tarefa simples para os bioinformáticos representar o seu conhecimento do domínio através das ontologias.

Atualmente existem algumas ferramentas de edição de ontologias que permitem ao utilizador inserir um ficheiro referente a uma ontologia e criar novos dados de acordo com este ficheiro, como por exemplo *Protégé* [3]. Contudo, não temos conhecimento da existência de uma ferramenta que também permita a transformação de dados semiestruturados.

Desta forma, de modo a ajudar os utilizadores – como por exemplo, os biólogos - desenvolveremos uma aplicação que tenha uma interface intuitiva que permita esta transformação de dados semiestruturados em dados anotados com ontologias definidas em OWL. Nesta interface também teria a possibilidade de anotar valores aos vários conceitos da ontologia ou apenas editar os existentes.

1. **Arquitetura e Requisitos**

A figura 1 representa a arquitetura da aplicação. Esta será desenvolvida de modo a que o utilizador (*User*) insira um ficheiro com definição de uma ontologia (*Ontology File*) e, opcionalmente, um segundo ficheiro (*Real Case File*). Estes ficheiros irão ser submetidos a uma aplicação externa designada Chaos Pop[[1]](#footnote-1) através do módulo *FileUploader*. De seguida irá ser gerada uma interface gráfica onde o user poderá anotar valores aos vários conceitos presentes no *Ontology File* ou anotar os conceitos do *Real Case File* com os termos do *Ontology File*. No final deste processo, é gerado um novo ficheiro OWL que contém os dados descritos de acordo com *Ontology File*. Iremos também dar a opção ao user de guardar os ficheiros de input e output numa base de dados remota.

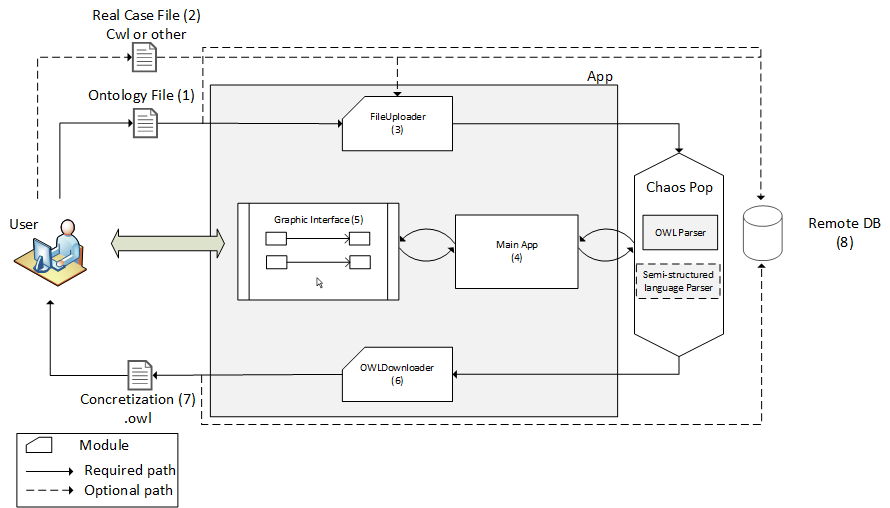


Figura 1 - Arquitetura da aplicação: fluxo de execução

Desta forma, este projeto tem como requisitos obrigatórios:

* Realizar uma versão remota da aplicação descrita anteriormente em tecnologia *Node.js*. Nesta será permitida a inserção de ficheiros CWL [4] (JSON) ou XML para *Real Case File* e irá existir a persistência dos ficheiros de *input* e de *output* numa base de dados remota documental.
* Realizar uma versão local utilizando a tecnologia *Electron* [[2]](#footnote-2) onde o ficheiro de *output* será salvo em disco.
* Definir em OWL a descrição de ferramentas bioinformáticas, previamente definidas em CWL.

Os requisitos seguintes só serão realizados como sejam concluídos com sucessos os obrigatórios:

* Definir em OWL *workflows,* previamente definidos em CWL.
* Adicionar ao módulo externo Chaos Popum *parser* para CWL descrito em YAML.
* Criar um sistema de autorização/autenticação para a versão Web.

1. **Calendarização**

Os termos utilizados nesta calendarização correspondem ao da Figura 1.

|  |  |  |
| --- | --- | --- |
| Data de início | Semana | Descrição |
| 19/02/2018 | 1-2 | - Compreensão da necessidade da ferramenta nos dias atuais  - Estudo do Chaos Pop |
| 05/03/2018 | 3-4 | - Estudo da ferramenta Electron  - Desenvolvimento da proposta |
| 19/03/2018 | 5 | - Entrega da proposta do projeto  - Utilização do Chaos Pop em alguns exemplos |
| 26/03/2018 | 6-7 | - Desenvolvimento do módulo *FileUploader* e *OWLDownloader* |
| 09/04/2018 | 8-10 | - Realização do esqueleto da *Main App*  - Desenvolvimento da *Graphic Interface* |
| 30/04/2018 | 11 | - Apresentação individual e entrega do relatório de progresso |
| 07/05/2018 | 12-13 | - Definição da descrição de ferramentas em OWL |
| 21/05/2018 | 14 | - Desenvolvimento da aplicação *desktop*  - Criação do cartaz |
| 28/05/2018 | 15 - 17 | - Entrega do cartaz e da versão beta  - Otimização dos módulos |
| 18/06/2018 | 18 | - Testes de escalabilidade |
| 25/06/2018 | 19-21 | - Finalização do relatório e entrega da versão final |

O relatório, os testes unitários e a documentação serão atualizados constantemente ao longo das 20 semanas.

|  |  |
| --- | --- |
| Marcos / Entregas | Data limite |
| Proposta do projeto | 19/03/2018 |
| Relatório de progresso e apresentação individual | 30/04/2018 |
| Cartaz e versão beta | 28/05/2018 |
| Versão final e documentação | 14/07/2018 |

# Referências

|  |  |
| --- | --- |
| [1] | “W3C,” [Online]. Available: https://www.w3.org/standards/semanticweb/ontology. [Acedido em 19 03 2018]. |
| [2] | “W3C,” [Online]. Available: https://www.w3.org/OWL/. [Acedido em 09 03 2018]. |
| [3] | “Protege,” [Online]. Available: https://protege.stanford.edu/. [Acedido em 15 03 2018]. |
| [4] | “Common Workflow Language,” [Online]. Available: http://www.commonwl.org/draft-3/UserGuide.html;. [Acedido em 09 03 2018]. |

**Bibliografia**

Jamie Taylor, Colin Evans, Toby Segaran. (2009). Programming the Semantic Web.

Jim R. Wilson. (2013). Node.js the Right Way: Practical, Server-side JavaScript that Scales.

1. A API da aplicação Chaos Pop ainda está em desenvolvimento pelo que iremos explicar melhor o seu funcionamento no relatório [↑](#footnote-ref-1)
2. https://electronjs.org/docs/tutorial/about [↑](#footnote-ref-2)