 **INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

**HYBRID ONTOLOGY MAPPING INTERFACE**

**H.O.M.I.**

**Projeto Final de Curso**

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

**Manual do utilizador**

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)

Julho de 2018

Conteúdo

[1 Instalação 4](#_Toc518410528)

[1.1 Requisitos 4](#_Toc518410529)

[1.1.1 Clone do repositório 4](#_Toc518410530)

[1.1.2 Instalação do node.js 4](#_Toc518410531)

[1.1.3 Instalação do MongoDb 4](#_Toc518410532)

[1.2 Executar o H.O.M.I 4](#_Toc518410533)

[2 Utilização 5](#_Toc518410534)

[2.1 Populate with data 5](#_Toc518410535)

[2.2 Populate without data 9](#_Toc518410536)

# Instalação

Não existe um ficheiro de instalação que permita instalar a versão desktop da aplicação, porém é possível usufruir dessa vertente com base em alguns requisitos e passos a serem seguidos.

## Requisitos

Primeiramente, é necessário realizar o *clone* do repositório *github* que contém o código da aplicação.

A seguir, é preciso instalar as seguintes ferramentas:

* *Node.js*
* *MongoDb*

Ambos os ficheiros de instalação estão disponíveis no repositório do projeto.

### 1.1.1 Clone do repositório

Para realizar o clone do repositório abra uma janela de comando numa diretoria da sua escolha e execute o seguinte comando:

//TODO: mudar para outro repositório publico onde tenha apenas a aplicação e os ficheiros de instalação do mongo e node

git clone <https://github.com/anabaptista/pfc.git>

Após isto, já tem acesso aos ficheiros de instalação mencionados acima.

### 1.1.2 Instalação do node.js

//TODO: testar num pc sem que tenha instalado

### 1.1.3 Instalação do MongoDb

//TODO: testar num pc sem que tenha instalado

## 1.2 Executar o H.O.M.I

Depois de realizar todos os passos de instalação anteriores, já pode executar a versão desktop da aplicação H.O.M.I. Para isto, navegue dentro da diretoria onde realizou o *clone* do repositório para a seguinte pasta:

dir …./reponame/Aplicação

Abra a linha de comandos nesta pasta e execute o seguinte comando:

npm start

Uma aplicação desktop irá ser aberta e já poderá navegar pela interface da H.O.M.I.

# Utilização

A *user interface* visível na página inicial é a apresentada abaixo:

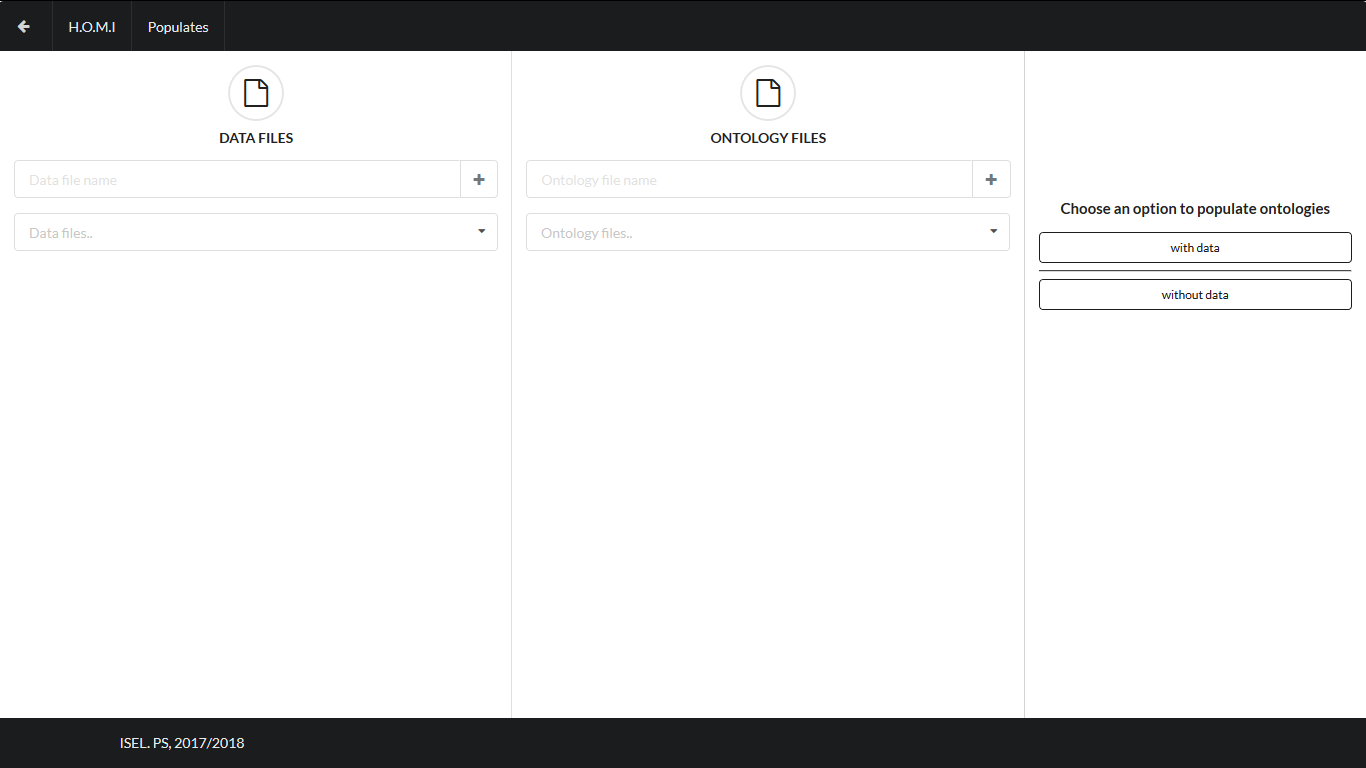


Figura . H.O.M.I home page

É nesta página que a interação começa sendo possível a inserção de ficheiros do tipo DataFile e OntologyFile, bem como escolher alguns dos ficheiros existentes para realizar um *Populate* com (*with data*) ou sem (*without data*) dados semi-estruturados. Também é possível ver todos os *populates* existentes.

## 2.1 Populate with data

Para realizar um *populate* deste tipo é necessário a seleção de um ou mais ficheiros de OntologyFile e DataFile (1) e em seguida realizar um pedido para *Populate with data* (2).

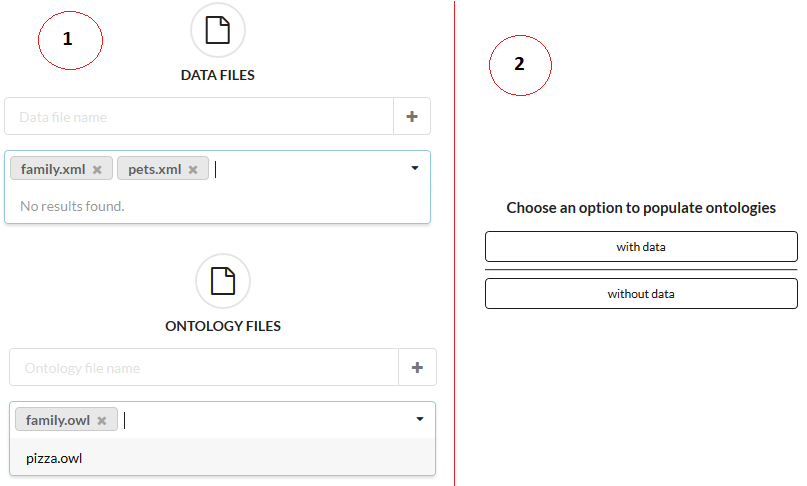


Figura . Populate with data

A página que será apresentada após o pedido, será a indicada na figura a seguir. Nesta página poderá criar *IndividualMapping’s*. Este representa o mapeamento de um determinado *node* da árvore (3), na qual será utilizado para mapear todos os *nodes* com aquela *tag* e não apenas aquele *node* em específico.

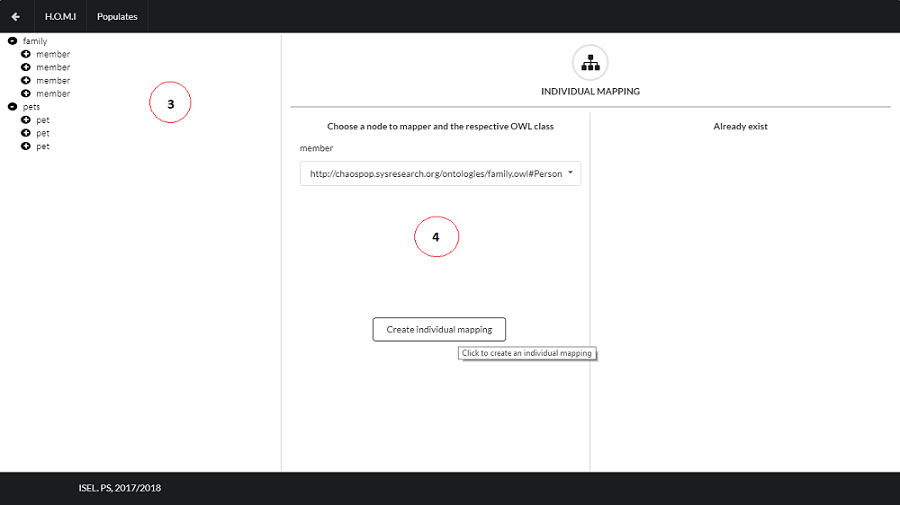


Figura . Populate page

A criação de um *IndividualMapping* inicia-se com a escolha do *node* e a *OWL class* referentes ao *IndividualMapping* que deseja criar. (4)

Após a criação do *IndividualMapping*, pode começar a mapear as propriedades do mesmo, sendo estas do tipo *data*, *object* e *annotation*, assim como atribuir um *IndividualName* (5) ao individuo criado. Como no exemplo acima foi selecionado o *node member* para mapeá-lo como *Person* (4), sendo assim o mapeamento das *properties* deste só pode ser realizado a partir dos *children* de *member*. (6). As regras de mapeamento para cada tipo de *property* são:

* Para mapear uma *annotation property* é necessário selecionar o *node* (ou os *nodes*, podem ser selecionados mais que um) que serão associados a este mapeamento, bem como o tipo da anotação que está a ser feita (7). São suportadas as seguintes anotações: *Label, Comment* e *SeeAlso.*
* Para mapear uma *data property* é necessário selecionar o *node* (ou os *nodes*, podem ser selecionados mais que um) que quer associar a este mapeamento bem como a *OWL* *data* IRI e o tipo desta propriedade (8). São aceites os tipos *Integer, Float, Double, String* e *Boolean.*
* Para mapear uma *object property* é necessário selecionar o *node* (apenas um) que será associado a essa propriedade e a *OWL object property* IRI a ser mapeada. (9).

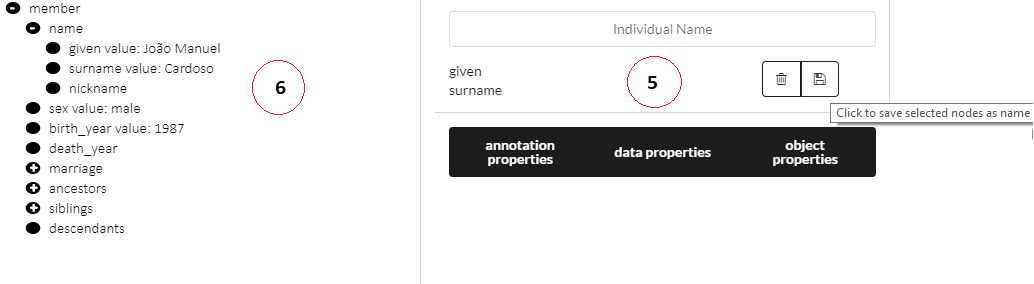


Figura . Individual Mapping nodes e IndividualName

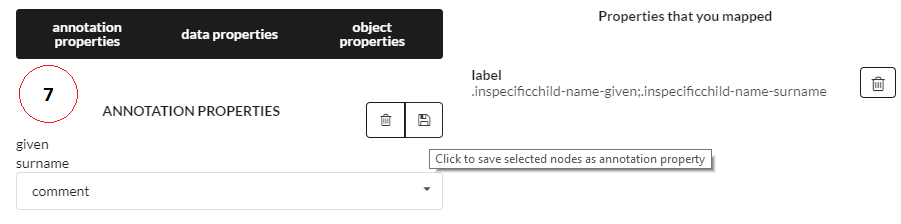


Figura . Individual Mapping annotation property

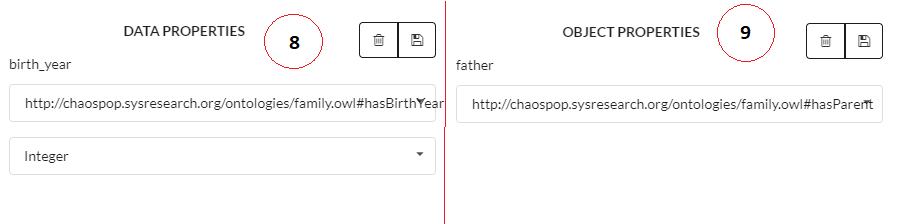


Figura . Individual Mapping data and object property

Depois de definir as propriedades que quer mapear, é necessário carregar no botão *SAVE* (10) presente na página do *IndividualMapping* em questão. Ao carregar neste botão, o *IndividualMapping* criado será enviado para o *Chaos Pop*.

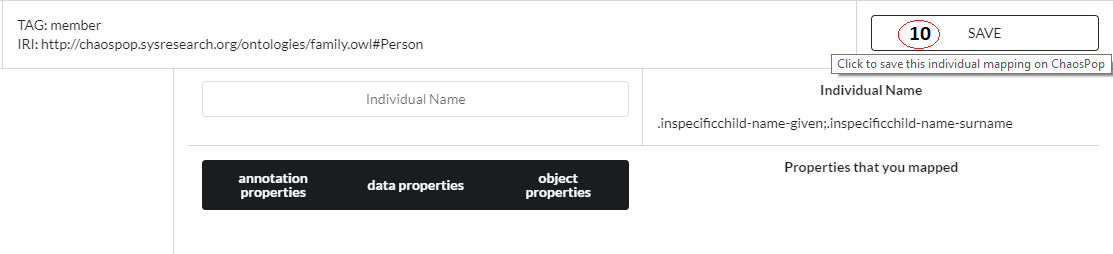


Figura . Botão SAVE

Quando pretender gerar o ficheiro de *output* dos dados mapeados é necessário primeiro criar um *Mapping*. A criação desteé possível clicando na barra de navegação em *Populates* (11), assim aparecerá a lista de *populates* criados e será possível clicar na opção *Create Mapping* do *populate* desejado (12).

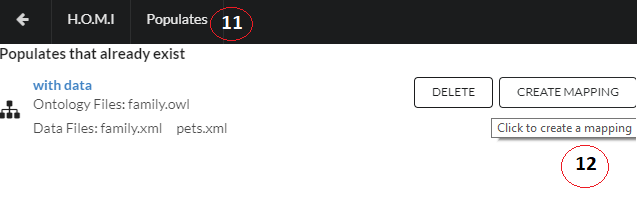


Figura . Lista de populates e criar mapping

Após clicar em *Create Mapping* é necessário selecionar quais *Individual Mapping’s* mapeados presentes na lista que deseja atribuir ao *mapping* que será criado (13). Na lista apenas aparece os *Individual Mapping’s* que foram submetidos ao *ChaosPop*, ou seja, que clicou no botão *save* (10). Também é preciso escolher qual o nome que pretende atribuir ao ficheiro de *output* e a seguir clicar em *submit* (14)*.*

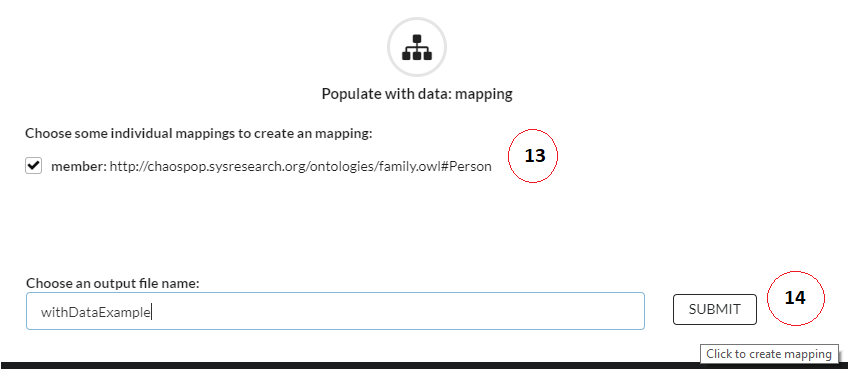


Figura . Seleção de individual mapping's e escolha do nome de ficheiro de output

//TODO: output file (falta o download e no entanto pode mudar)

## 2.2 Populate without data

Um *populate* deste género é realizado apenas com ficheiros de ontologia. Logo, o primeiro passo para popular uma ontologia é escolher os ficheiros desejados (1) e a seguir optar pelo *populate without data* (2)*.*

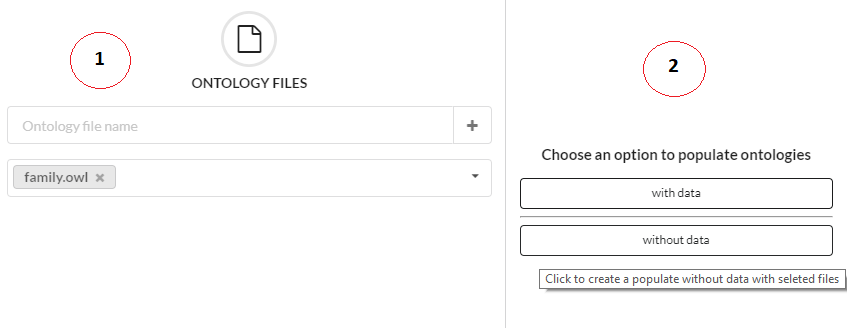


Figura . Iniciar um populate without data

Após a criação do *populate* pode se criar indivíduos escolhendo um nome para tal e a respetiva classe da ontologia (3).

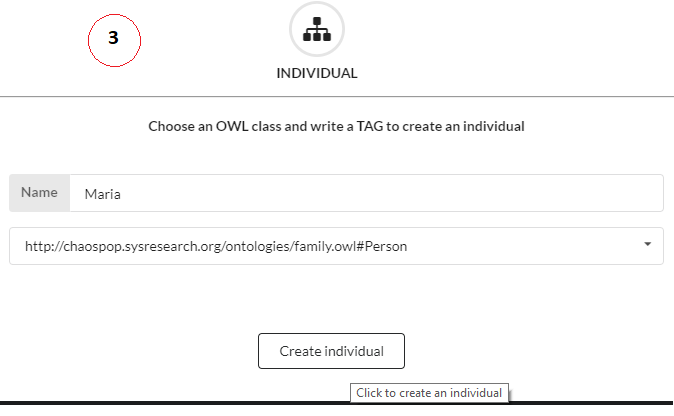


Figura . Criação de indivíduos

Assim como no *populate with data,* é possível adicionar propriedades dos tipos *annotation*, *data* e *object* aos indivíduos criados. As propriedades são caracterizadas por:

* *Annotation property*: um tipo de anotação (*Label, Comment* ou *SeeAlso*) e um valor para atribuir a anotação selecionada (4).
* *Data property*: uma classe da ontologia referente a *data property*, um valor para atribuir a essa propriedade e qual o tipo do valor atribuído (*Integer, Float, Double, String* e *Boolean*) (5).
* *Object property*: uma classe da ontologia refente a *object property* e um indivíduo anteriormente criado para associar a essa (6).

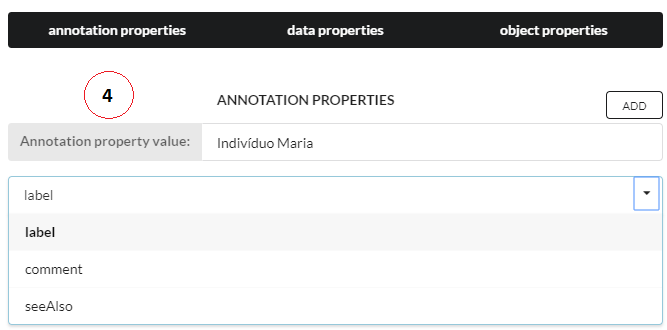


Figura . Inserção de annotation property

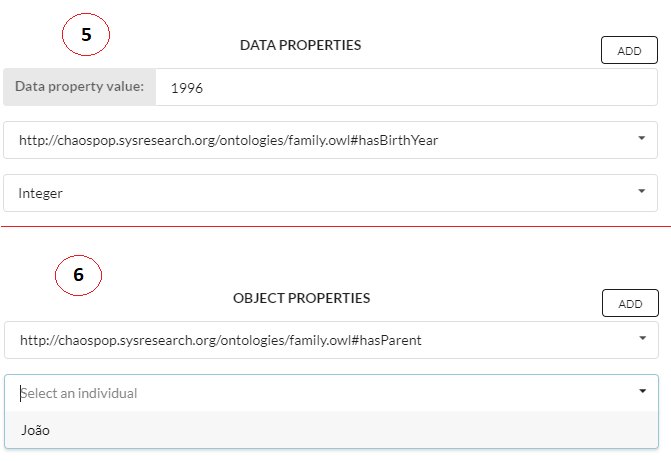


Figura . Inserção de data e object properties

Quando desejar obter o ficheiro de *output* é necessário navegar até *Populates* (7) e realizar o *mapping* dos indivíduos criados (8).

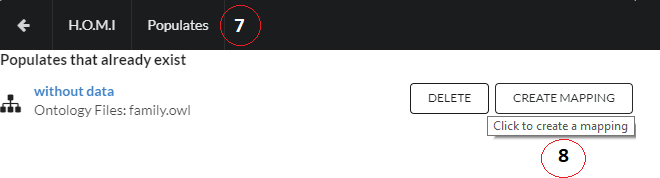


Figura . Selecionar populate para criar mapping

Para criar o *mapping* do *populate* em questão é preciso primeiro selecionar quais indivíduos estarão presentes neste *mapping* (9) e qual o nome deseja atribuir ao ficheiro de *output* (10) para depois clicar em *submit* (11).

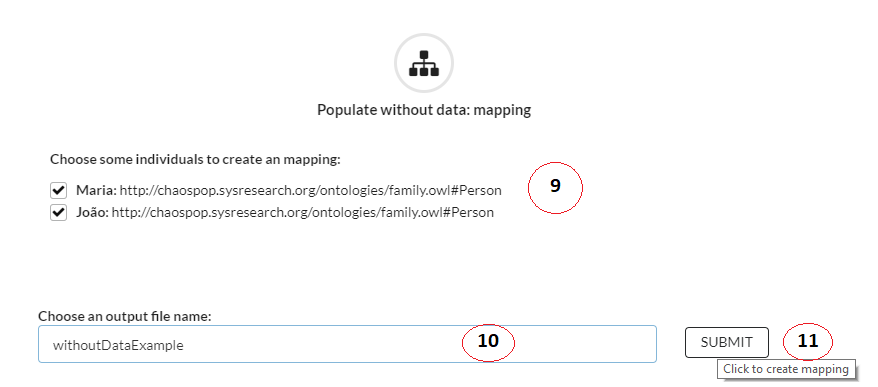


Figura . Criação do mapping da ontologia populada

//TODO: fazer download e mostrar ficheiro