



**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

Frederico Moreira Ribeiro

**Especificação de uma Ontologia  
para Genealogia**

December 2017



**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

Frederico Moreira Ribeiro

## **Especificação de uma Ontologia para Genealogia**

Master dissertation

Master Degree in Computer Science

Dissertation supervised by

**José Carlos Leite Ramalho**

December 2017

---

## AGRADECIMENTOS

---

Quero agradecer à minha família e amigos pelo suporte que me deram e à minha namorada que sempre me apoiou porque sem eles não teria conseguido terminar mais esta etapa.

Agradeço também ao Professor José Carlos Ramalho que me orientou ao longo desta dissertação e que não desistiu de mim mesmo nos momentos em que parecia impossível chegar a bom porto.

---

## ABSTRACT

---

The subject of this dissertation is the specification of an ontology of genealogy, the main objective being to develop an ontology.

The ontology will be able to support the description of the genealogy of a group of individuals, from personal data going through the relations of kinship to significant events of a person's life, and may also associate photographs or documentary records that are of interest.

In addition, we intend to develop a Web application that supports all ontology management and maintenance operations.

This type of application can be used as a hobby or exercise of simple curiosity by the family past or as support of family history for medical, genetic, among others.

---

## RESUMO

---

O tema desta dissertação é a especificação de uma ontologia de genealogia, sendo que o principal objetivo é desenvolver uma ontologia.

A ontologia será capaz de suportar a descrição da genealogia de um conjunto de indivíduos, desde dados pessoais passando pelas relações de parentesco até eventos significativos da vida de uma pessoa, podendo associar fotografias ou registos documentais que sejam de interesse.

Para além disso, pretendia-se desenvolver uma aplicação Web que suportasse todas as operações de gestão e manutenção da ontologia.

Este tipo de aplicação pode ser usada como passatempo ou exercício de simples curiosidade pelo passado familiar ou como suporte de histórico familiar para fins médicos, genéticos, entre outros.

---

## CONTEÚDO

---

1	INTRODUÇÃO	1
2	ESTADO DA ARTE	3
2.1	Ontologia	3
2.1.1	Elementos de uma Ontologia	4
2.1.2	Tipos de Ontologia	4
2.1.3	RDF	6
2.1.4	OWL	7
2.1.5	Metodologias de Desenvolvimento	9
2.2	Genealogia	12
2.2.1	Árvore Genealógica	12
2.2.2	Parentesco	13
2.3	Plataformas existentes	14
2.3.1	MyHeritage	14
2.3.2	FamilySearch	15
2.3.3	Outros Exemplos	16
2.4	GEDCOM	17
2.5	GraphDB	19
2.6	SPARQL	20
3	MODELAÇÃO E CONSTRUÇÃO DA ONTOLOGIA	21
3.1	Desenvolvimento da Ontologia	21
3.1.1	Determinar Contexto	21
3.1.2	Considerar Reutilização	22
3.1.3	Enumerar Conceitos	22
3.1.4	Definir Classes	22
3.1.5	Definir Propriedades	25
3.1.6	Criar Instâncias	37
4	IMPLEMENTAÇÃO	38
4.1	Arquitetura Geral	38
4.2	Repositório para a Ontologia	39
4.2.1	Importar Ontologia no GraphDB	39
4.2.2	Utilidades do GraphDB	41
4.3	XAMPP	43
4.4	Comunicação entre a Aplicação Web e o GraphDB	44

4.4.1	Sparqllib	44
4.5	Aplicação Web	45
4.5.1	HTML	45
4.5.2	PHP	46
4.5.3	JavaScript	46
4.5.4	W3.CSS	47
4.5.5	Tecnologias Usadas	47
4.5.6	Estrutura do Código/Pastas	48
4.5.7	Template Usado - W3.CSS	49
4.5.8	Operações CRUD - SPARQL	50
5	APLICAÇÃO - EXEMPLOS	64
5.1	Interface Principal	64
5.2	Funcionalidades relativas a Pessoas	66
5.2.1	Lista de Pessoas	66
5.2.2	Detalhes da Pessoa	67
5.2.3	Adicionar	68
5.2.4	Editar	69
5.2.5	Eliminar	71
5.3	Funcionalidades relativas a Eventos	72
5.3.1	Lista de Eventos	72
5.3.2	Adicionar	73
5.3.3	Eliminar	74
5.4	Funcionalidades relativas a Documentos	75
5.4.1	Lista de Documentos	76
5.4.2	Adicionar	77
5.4.3	Eliminar	78
5.5	Sobre	79
6	CONCLUSÃO	80
6.1	Conclusões	80
6.2	Trabalho Futuro	82

---

## LISTA DE FIGURAS

---

Figura 1	Tipos de ontologias separadas pela abordagem	5
Figura 2	Ontology Development 101	9
Figura 3	Methontology	10
Figura 4	Parentescos	13
Figura 5	Interface MyHeritage	14
Figura 6	Modelo de Dados - parte 1	17
Figura 7	Modelo de Dados - parte 2	18
Figura 8	Exemplo de um grafo de uma rede social	19
Figura 9	Classes da Ontologia	23
Figura 10	Classes e Subclasses da Ontologia - Aplicação Web	24
Figura 11	Diagrama de Propriedades da Classe - Person	25
Figura 12	Diagrama de Propriedades da Classe - Event	26
Figura 13	Diagrama de Propriedades da Classe - Document	27
Figura 14	Diagrama de Propriedades das Classes - Family e FamilyRole	27
Figura 15	Tabela de Relações - Parte 1	29
Figura 16	Tabela de Relações - Parte 2	30
Figura 17	Tabela de Relações - Parte 3	31
Figura 18	Esquema "Inverse Of" da inferência para Avós	32
Figura 19	Esquema "SuperProperty Of (Chain)" da inferência para Avós	33
Figura 20	Esquema "SuperProperty Of (Chain)" da inferência para Tios	33
Figura 21	Esquema "SuperProperty Of (Chain)" da inferência para Sogros	35
Figura 22	Esquema "SuperProperty Of (Chain)" da inferência para Cunhados	36
Figura 23	Arquitetura	38
Figura 24	Interface do GraphDB - Repositórios	40
Figura 25	Interface do GraphDB - Classes	40
Figura 26	Interface do GraphDB - SPARQL	41
Figura 27	Interface do Ontotext Cognitive Cloud	42
Figura 28	Interface do XAMPP	43
Figura 29	Estrutura de uma página HTML	45
Figura 30	Estrutura das Pastas	48
Figura 31	Template Usado	49
Figura 32	Página Principal	64
Figura 33	Menu de Atalhos	65



Figura 34	Listagem de Pessoas da Família	66
Figura 35	Dados da Pessoa	67
Figura 36	Pais da Pessoa	67
Figura 37	Inserir Pessoa	68
Figura 38	Editar Pessoa	69
Figura 39	Adicionar Relações de Parentesco	70
Figura 40	Mensagem associada à impossibilidade de se eliminar uma pessoa	71
Figura 41	Lista de Eventos	72
Figura 42	Lista de Emigrações	72
Figura 43	Adicionar Eventos	73
Figura 44	Eliminar Evento	74
Figura 45	Pastas para os Documentos	75
Figura 46	Lista Documentos	76
Figura 47	Lista de Fotografias	76
Figura 48	Adicionar Documento	77
Figura 49	Eliminar Documento	78
Figura 50	Sobre	79

---

## INTRODUÇÃO

---

Esta dissertação enquadra-se no 2º ano do Mestrado em Engenharia Informática da Universidade do Minho.

O passado da nossa família e as pessoas que fazem parte dela é algo que pertence à história de cada um de nós e por isso surge o interesse em criar a árvore genealógica da sua família tanto por motivos pessoais e de simples curiosidade como por motivos profissionais.

Este interesse levou a que já fossem criados alguns websites próprios para a elaboração de árvores genealógicas que podem depois ser interligadas com as árvores de outros utilizadores caso haja alguma pessoa que seja coincidente em várias árvores, facilitando assim a construção das mesmas.

A árvore genealógica de uma pessoa é um domínio bem definido para a criação de uma ontologia que servirá de base para uma aplicação web.

Os objetivos desta tese de mestrado são:

- Desenvolver uma ontologia capaz de suportar a descrição da genealogia de um conjunto de indivíduos: dados pessoais, relações de parentesco, eventos como matrimónio, nascimento, óbito, aniversários e outros, associação de representações digitais como fotos, registos documentais/oficiais de eventos ou ações e outros;
- Desenvolver uma aplicação Web que suporte todas as operações de gestão e manutenção da ontologia bem como a sua exploração;

Neste documento irão estar presentes alguns conceitos teóricos associados ao domínio definido para se obter um enquadramento sobre o mesmo.

Haverá também um breve resumo das tecnologias que serão usadas no desenvolvimento de todos os componentes do sistema pretendido.

Por fim serão apresentados todos os procedimentos desenvolvidos ao longo destes meses de trabalho complementados por exemplos da aplicação funcional.

---

## ESTADO DA ARTE

---

### 2.1 ONTOLOGIA

No contexto das ciências da computação, uma ontologia é uma conceptualização no sentido em que captura/representa um modelo abstrato e cognitivo de um domínio de conhecimento, identificando os conceitos e as características desse mesmo domínio.

Essa conceptualização deve ser partilhada de modo a representar o conhecimento consensual de um grupo e não apenas o conhecimento privado de um indivíduo ou organização. Da mesma forma, uma ontologia é também vista como parcial no sentido de refletir a impossibilidade de capturar de forma completa todo o conhecimento do domínio em causa.

Podemos também falar de uma ontologia como uma especificação em que através de uma linguagem de representação de ontologias (ex: RDFS, OWL) a mesma torna-se tangível (isto é, tem uma existência física). Esta especificação deve ser formal para que se torne legível por computador (machine-readable) e explícita no sentido em que as entidades da ontologia estão claramente definidas, distintas e inter-relacionadas entre si.

Independentemente do domínio de conhecimento em causa, as ontologias são maioritariamente usadas em processos relacionados com a aquisição e representação de informação, bem como com a partilha e integração de informação entre sistemas.

Uma ontologia é constituída por um vocabulário com os seguintes elementos:

- Classes;
- Atributos;
- Relações;
- Indivíduos/Objetos.

[Roc16a]

### 2.1.1 Elementos de uma Ontologia

#### *Classes*

Normalmente organizadas em taxonomias, as classes representam algum tipo de interação da ontologia com um determinado domínio. Elas podem conter indivíduos, outras classes, ou uma combinação de ambos.

#### *Atributos*

Cada atributo é utilizado para armazenar informação que é específica para o objeto associado a ele.

#### *Relações*

Representam o tipo de interação entre os elementos do domínio (classes) e os indivíduos.

#### *Indivíduos/Objetos*

São utilizados para representar elementos específicos, isto é, os próprios dados da ontologia.

### 2.1.2 Tipos de Ontologia

Existem vários tipos de ontologias que se podem agrupar, segundo a sua abordagem em 5 grupos diferentes. Na tabela apresentada abaixo encontram-se listados os grupos que dividem a classificação das ontologias de forma a entender melhor a variedade existente.

Cada grupo contém em si diferentes tipos de classificação de ontologias.

- Quanto à Função;
- Quanto ao Grau de Formalismo;
- Quanto à Aplicação;
- Quanto à Estrutura;
- Quanto ao Conteúdo.

Abordagem	Classificação	Descrição
<b>Quanto à função</b> Mizoguchi, Vanwelkenhuysen & Ikeda (1995)	Ontologias de domínio	Reutilizáveis no domínio, fornecem vocabulário sobre conceitos, seus relacionamentos, sobre atividades e regras que os governam.
	Ontologias de tarefa	Fornecem um vocabulário sistematizado de termos, especificando tarefas que podem ou não estar no mesmo domínio.
	Ontologias gerais	Incluem um vocabulário relacionado a coisas, eventos, tempo, espaço, casualidade, comportamento, funções etc.
<b>Quanto ao grau de formalismo</b> Uschold & Gruninger (1996)	Ontologias altamente informais	Expressa livremente em linguagem natural.
	Ontologias semi-informais	Expressa em linguagem natural de forma restrita e estruturada.
	Ontologias semiformais	Expressa em uma linguagem artificial definida formalmente.
	Ontologia rigorosamente formal	Os termos são definidos com semântica formal, teoremas e provas.
<b>Quanto à aplicação</b> Jasper & Uschold (1999)	Ontologias de autoria neutra	Um aplicativo é escrito em uma única língua e depois convertido para uso em diversos sistemas, reutilizando-se as informações.
	Ontologias como especificação	Cria-se uma ontologia para um domínio, a qual é usada para documentação e manutenção no desenvolvimento de <i>softwares</i> .
	Ontologias de acesso comum à informação	Quando o vocabulário é inacessível, a ontologia torna a informação inteligível, proporcionando conhecimento compartilhado dos termos.
<b>Quanto à estrutura</b> Haav & Lubi (2001)	Ontologias de alto nível	Descrevem conceitos gerais relacionados a todos os elementos da ontologia (espaço, tempo, matéria, objeto, evento, ação etc.) os quais são independentes do problema ou domínio.
	Ontologias de domínio	Descrevem o vocabulário relacionado a um domínio, como, por exemplo, medicina ou automóveis.
	Ontologias de tarefa	Descrevem uma tarefa ou atividade, como, por exemplo, diagnósticos ou compras, mediante inserção de termos especializados na ontologia.
<b>Quanto ao conteúdo</b> Van-Heijst, Schreiber & Wielinga (2002)	Ontologias terminológicas	Especificam termos que serão usados para representar o conhecimento em um domínio (por exemplo, os léxicos).
	Ontologias de informação	Especificam a estrutura de registros de bancos de dados (por exemplo, os esquemas de bancos de dados).
	Ontologias de modelagem do conhecimento	Especificam conceitualizações do conhecimento, têm uma estrutura interna semanticamente rica e são refinadas para uso no domínio do conhecimento que descrevem.
	Ontologias de aplicação	Contêm as definições necessárias para modelar o conhecimento em uma aplicação.
	Ontologias de domínio	Expressam conceitualizações que são específicas para um determinado domínio do conhecimento.
	Ontologias genéricas	Similares às ontologias de domínio, mas os conceitos que as definem são considerados genéricos e comuns a vários campos.
Ontologias de representação	Explicam as conceitualizações que estão por trás dos formalismos de representação do conhecimento.	

Figura 1: Tipos de ontologias separadas pela abordagem  
[AB03]

Na tabela acima podemos ver uma pequena descrição de cada tipo de ontologia. Cada ontologia pode ter mais de uma destas classificações, por exemplo, pode ser uma ontologia altamente informal e geral ao mesmo tempo.

A ontologia criada podemos classificá-la como sendo de domínio transversalmente às classificações de:

- quanto à função,
- quanto à estrutura,
- quanto ao conteúdo.

### 2.1.3 RDF

RDF [RDF14] é uma sigla que provém de Resource Description Framework e é um modelo de dados abstrato para representação de informação na Web. Contrariamente aos modelos de dados mais tradicionais, como é o caso do modelo aplicado nas base de dados relacionais ou do modelo em árvore típico dos documentos XML, a informação representada em RDF constitui um grafo.

Dois fatores que distinguem o RDF dos outros modelos referidos anteriormente são:

- A sua flexibilidade inata para representar novos dados.
- A sua capacidade para representar de igual forma tanto os dados como o esquema subjacente a esses dados.

Os grafos RDF são constituídos por dois tipos de elementos: Nós e Arcos.

Em RDF, um triplo ou facto (statement) é constituído por um sujeito, predicado e objeto. O sujeito corresponde sempre a um recurso representado pelo seu IRI (Internationalized Resource Identifier), o predicado corresponde sempre a uma propriedade representada também pelo seu IRI e o objeto corresponde a um recurso (representado pelo seu IRI) ou a um literal.

Um exemplo para ajudar na percepção da estrutura dos grafos referida no parágrafo anterior: 'Pessoa1' - 'temNome' - 'João' (sujeito - predicado - objeto). [Roc15a]

#### 2.1.4 OWL

O OWL [OWL13] (Web Ontology Language), tal como o RDF, foi criado para ser usado por aplicativos que precisam processar o conteúdo da informação em vez de apenas apresentar informações aos seres humanos.

OWL facilita a maior interpretabilidade da máquina do conteúdo da Web do que a suportada pelo XML, RDF e RDF Schema (RDF-S), fornecendo vocabulário adicional, juntamente com uma semântica formal.

Com OWL, além de se poder expressar a estrutura de conceitos e relacionamentos como acontece no RDF, é possível também descrever características especiais sobre os conceitos e sobre os relacionamentos através dos axiomas lógicos.

Comparativamente a outras linguagens de esquema (exemplos: o XSD, o UML, linguagem de definição de esquema do SQL), o OWL apresenta essencialmente três vantagens:

- É mais expressivo, na medida em que permite expressar ideias muito complexas e subtis sobre os dados. Por exemplo, contrariamente ao UML onde a hierarquia de classes é estabelecida de forma estática, em OWL essa hierarquia pode ser estabelecida de forma dinâmica.
- É mais flexível, no sentido em que a especificação de um esquema em OWL corresponde à adição de novos triplos RDF e, portanto, é por natureza incremental. Logo, as aplicações assentes em OWL tendem a encarar alterações ao esquema de uma forma mais simples e natural do que as aplicações assentes nas linguagens de esquema mais tradicionais.
- É mais eficiente, na medida em que promove e suporta diversos tipos de inferência, o que permite, por exemplo, minimizar os dados que são explicitamente armazenados e a complexidade das consultas necessárias para recuperar esses mesmos dados. [Roc16b]

O OWL possui três formatos que vão aumentando de expressividade de uns para os outros, cada um deles é a extensão da anterior, tanto em relação ao que pode ser expressado, como em relação ao que pode ser concluído.

- OWL Lite - destina-se às ontologias que precisem apenas de descrição de restrições e hierarquia de classes simples.
- OWL DL - é usada para ontologias que precisem do máximo de expressividade, com completitude (todas as conclusões são garantidamente computáveis) e decidibilidade



(todas as computações terminarão num tempo finito) computacional. Ela inclui todas as construções da linguagem OWL, mas estas construções somente podem ser usadas sob certas restrições.

- OWL Full - destina-se às ontologias que precisem do máximo de expressividade e independência sintática de RDF, sem nenhuma garantia computacional. O OWL Full e o OWL DL suportam o mesmo conjunto de construções da linguagem OWL, embora com restrições um pouco diferentes. Enquanto a OWL DL impõe restrições sobre o uso de RDF e requer disjunção de classes, propriedades, indivíduos e valores de dados, o OWL Full permite misturar OWL com RDF Schema e não requer a disjunção de classes, propriedades, indivíduos e valores de dados. Isto é, uma classe pode ser ao mesmo tempo uma classe e um indivíduo.

A escolha do formato OWL que os desenvolvedores de ontologias devem usar vai depender sempre das necessidades da ontologia que pretendem construir.

### 2.1.5 Metodologias de Desenvolvimento

Existem metodologias para o desenvolvimento de ontologias que diferem em certos aspectos entre si, algumas delas serão descritas de seguida.

#### *Ontology Development 101*

Esta metodologia foi criada por pesquisadores da Universidade de Stanford, sendo a mais utilizada para a construção de ontologias nos dias de hoje. Uma das principais vantagens desta metodologia é a sua simplicidade o que justifica a sua maior utilização.

A metodologia consiste em sete passos iterativos simples que servem de guia para os engenheiros de ontologias durante o processo de construção de uma ontologia. Os 7 passos são os apresentados na imagem abaixo. [IB14]

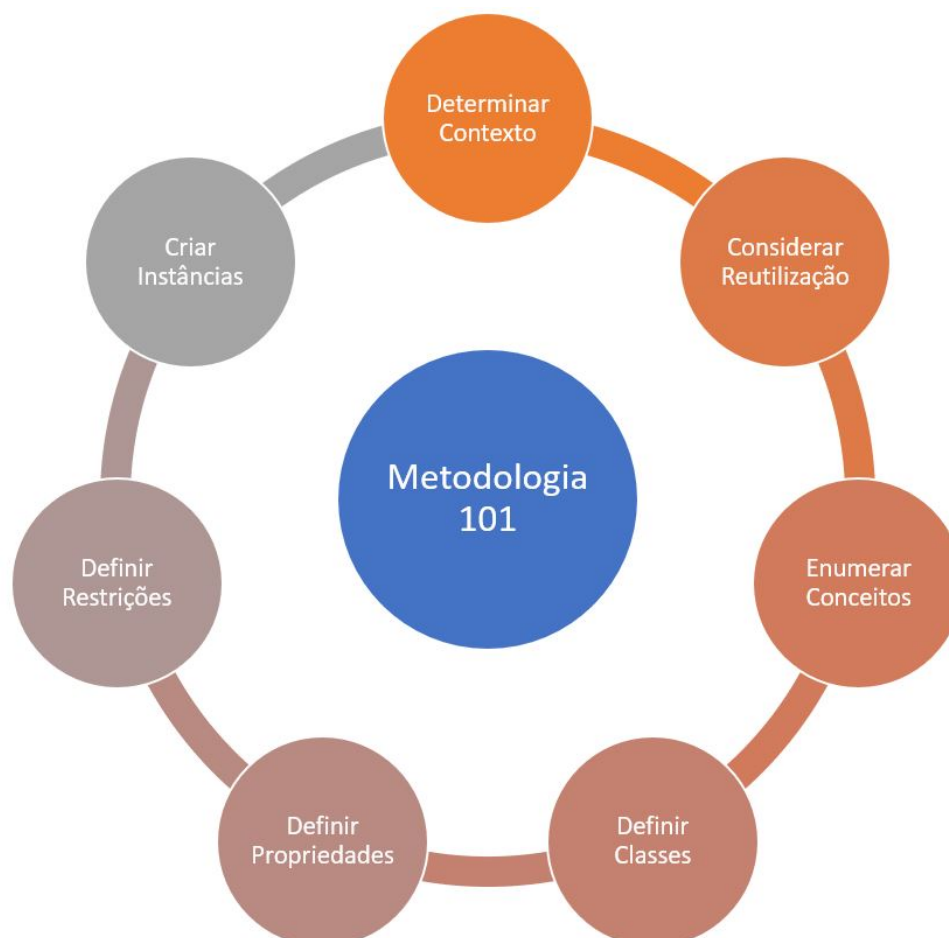


Figura 2: Ontology Development 101

### *On-to-knowledge*

On-to-Knowledge é uma metodologia de desenvolvimento de ontologias que resulta da cooperação entre várias entidades europeias, o seu objetivo é obviamente o de desenvolver ontologias, neste caso, ontologias para serem usadas em Sistemas de Gestão do Conhecimento. Esta metodologia encontra-se dividida em cinco fases:

- Estudo de viabilidade,
- Início da ontologia,
- Refinamento,
- Avaliação,
- Manutenção e Evolução. [RTSGo8]

### *Methontology*

Methontology é uma metodologia de desenvolvimento de ontologias idealizada por um grupo de pesquisa em Engenharia de Ontologias da Universidade Politécnica de Madrid, sendo fortemente influenciada por metodologias de Engenharia de Software e de Engenharia do Conhecimento. [RTSGo8]

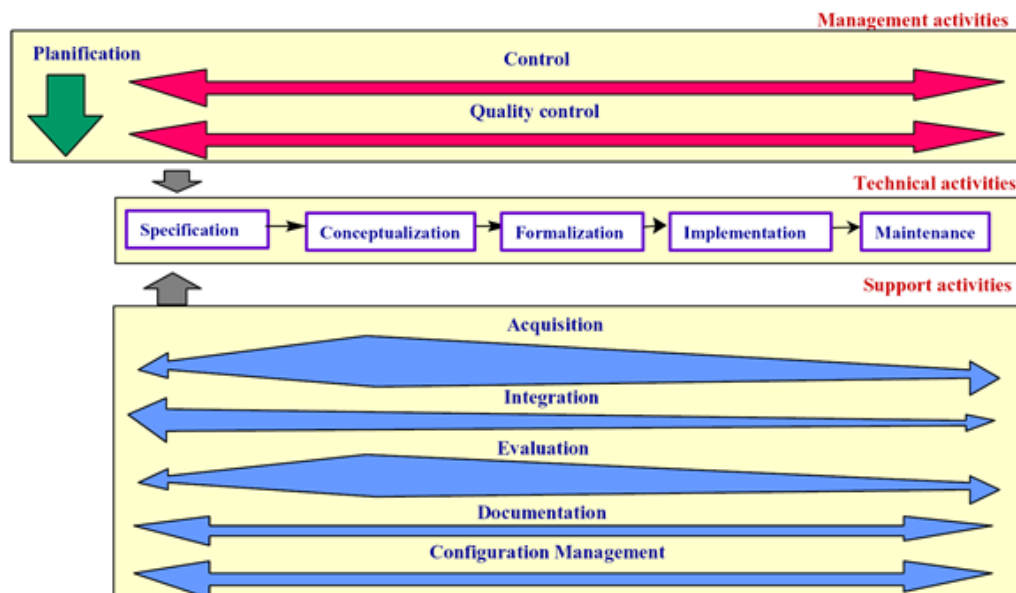


Figura 3: Methontology

*Enterprise*

Esta metodologia tem como base quatro fases. Sendo elas as seguintes:

- A identificação do propósito determina o nível de formalidade que a ontologia deve ser descrita,
- Na identificação do contexto uma especificação é produzida de acordo com o domínio que a ontologia precisa representar,
- A formalização é a criação do código, definições formais e axiomas relacionados à ontologia,
- A documentação formal é a fase onde a ontologia será documentada e as fases de identificação do contexto e formalização podem ser revistas. [MA07]

## 2.2 GENEALOGIA

Como se pode verificar no tema desta dissertação o domínio da ontologia é a genealogia, antes de mais a questão que se coloca é: O que é genealogia?

A genealogia é a ciência que tem por objetivo a pesquisa da origem e da filiação das famílias. Do latim *genealogia*, a genealogia é o conjunto de ascendentes e progenitores de uma pessoa.

A genealogia pode conhecer-se através daquilo que é contado pelos familiares ao longo da vida. Estas histórias são transmitidas de geração em geração e podem não ser exatas no que diz respeito aos dados mais específicos embora possam ser o ponto de partida para fazer estudos mais precisos.

Outra fonte documental da genealogia são os documentos impressos ou manuscritos. Ao contrário da memória familiar, estes registos oferecem exactidão na informação. Os registos de identificação civil, os documentos eclesiásticos (da Igreja) e os arquivos do Estado permitem conhecer a genealogia das pessoas. [Con16]

### 2.2.1 *Árvore Genealógica*

Com os dados que se recolhem dessas histórias familiares ou documentos impressos e manuscritos podemos construir a árvore genealógica da nossa família.

A árvore genealógica, traça o perfil das pessoas que fazem ou fizeram parte da família através de parentesco direto ou indireto. É um histórico que vai apresentar os antepassados da família, de maneira que tais conhecimentos proporcionem um esclarecimento do passado muitas vezes desconhecido.

Para os descendentes é como decifrar o mistério de um passado desconhecido, a grande chance de se conhecer avós, avôs, bisavós e bisavôs, entre outros parentes até então existentes só no imaginário das pessoas e nas histórias relatadas pelos familiares.

Normalmente quando se estrutura a árvore genealógica de uma família com poucas informações ela é representada por uma pequena árvore, quando os elementos são de maior proporção a sua feição torna-se mais complexa, pois os dados passam a ficar mais desarranjados, quando isso acontece a visualização melhora através de uma representação

gráfica, existindo já vários sistemas que ajudam nesta tarefa.

Uma árvore genealógica é de extrema importância para as famílias, pois por meio delas é possível ter conhecimento da ascendência familiar, da existência de títulos até então desconhecidos, sem dizer que futuramente, ela pode ser útil nos tratamentos de diversas anomalias e doenças genéticas. [Inf16]

2.2.2 Parentesco

A relação entre duas pessoas da mesma família é conhecida como parentesco havendo vários graus de parentesco para classificar essa relação dependendo de vários fatores. Ao longo da construção de uma árvore genealógica vamos descobrindo novas ligações familiares, umas tendo um parentesco mais próximo e outras com parentesco mais afastado.

Existem dois tipos de parentesco: um é o de sangue ou consanguíneo, o outro é o social ou de afinidade. Cada um destes grupos tem depois vários graus dependendo da proximidade do parentesco.

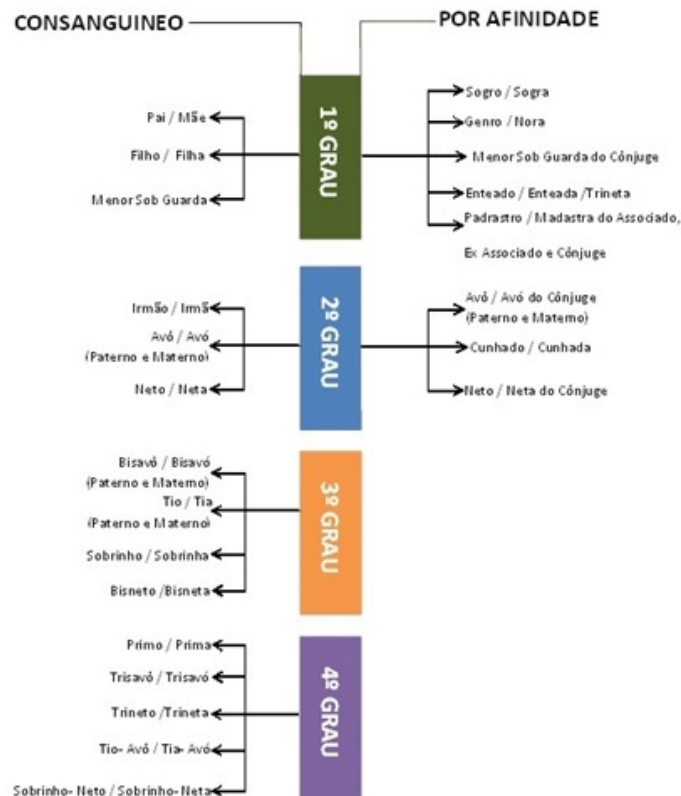


Figura 4: Parentescos

## 2.3 PLATAFORMAS EXISTENTES

Saber a história da sua família e os seus antepassados é algo que desperta a curiosidade das pessoas, sendo que a procura por estes dados tem aumentado fruto dessa curiosidade.

O número de ferramentas na área da genealogia tem aumentado o que facilita a obtenção de informações, juntando-se a isso o facto de que contactar familiares hoje em dia com recurso a tecnologia é muito acessível, temos assim um conjunto de circunstâncias que levam a que muitas pessoas procurem construir a sua árvore genealógica.

Esta dissertação vai partir da análise de sistemas comerciais já existentes, nas secções seguintes serão descritos alguns deles.

### 2.3.1 MyHeritage

MyHeritage [MyH16] é uma plataforma de genealogia online, que oferece serviços e produtos web, móveis e de software. Utilizadores da plataforma podem criar árvores genealógicas, fazer upload de fotos e visualizá-las, bem como pesquisar biliões de registos históricos globais, entre outras funções.

No final de 2015, a empresa disponibilizava o serviço em 42 idiomas e contava com cerca de 80 milhões de utilizadores no mundo todo, facto que comprova o grande número de pessoas com interesse nesta área.

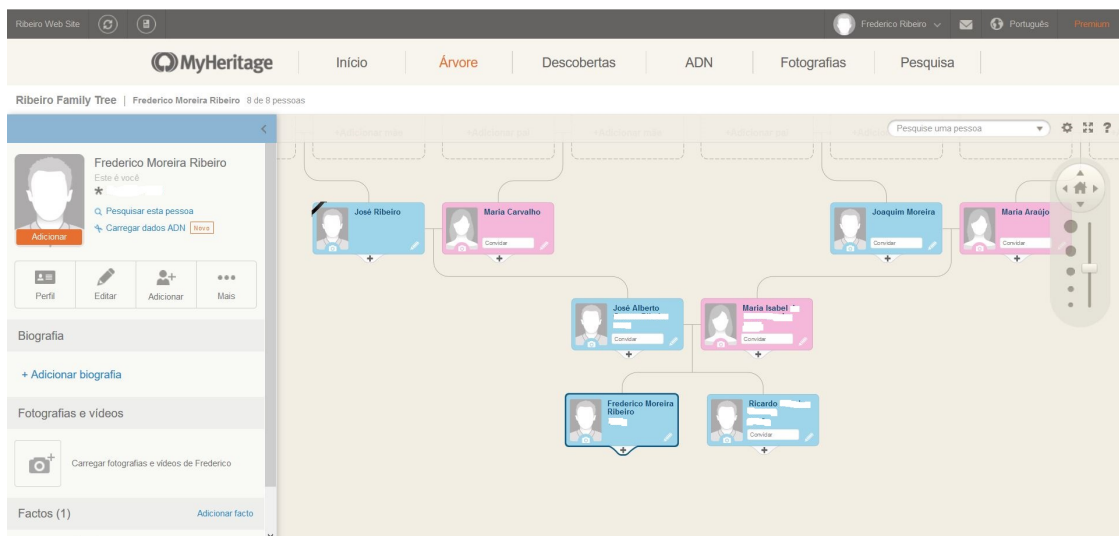


Figura 5: Interface MyHeritage

### 2.3.2 *FamilySearch*

FamilySearch (antiga Sociedade Genealógica de Utah) é uma organização de pesquisa genealógica mantida pela Igreja de Jesus Cristo dos Santos dos Últimos Dias e é a maior organização voltada para esse tipo de pesquisa no mundo.

O FamilySearch [Fam16a] mantém uma coleção de registros, recursos e serviços gratuitos destinados a ajudar as pessoas a aprenderem mais sobre a história da sua própria família. Todos os serviços prestados pela organização no seu website são gratuitos e disponíveis para o uso do público em geral. O FamilySearch presta ainda serviço a mais de 4800 centros de história da família em 134 países.

Para facilitar o armazenamento de dados genealógicos eles criaram um formato próprio para o efeito, denominado de GEDCOM, mais à frente este formato será descrito neste documento.



### 2.3.3 Outros Exemplos

Existem outras plataformas a maioria em Inglês mas algumas também em Português, sendo essas plataformas de vários níveis de complexidade. Umas apenas para construção da árvore outras com mais detalhe e mais funções além da construção da árvore. Segue uma lista de alguns exemplos:

- Geneanet (PT) - Criar árvore genealógica, importar e exportar ficheiro GEDCOM, cruzar árvores e projetos de levantamento genealógico são algumas das opções desta plataforma; [Gen16a]
- Geni (PT) - Funciona quase como uma rede social com o objetivo de criar a árvore genealógica com as relações que se vão estabelecendo, faz também parte do projeto MyHeritage; [Gen16b]
- Ancestry (ING) - É uma plataforma paga, de grandes dimensões em termos de registos históricos, com cerca de 16 biliões de registos; [Anc16]
- FamilyTreeNow (ING) - Plataforma com registos dos Censos dos Estados Unidos da América desde 1790 entre outros tipos de registos; [Fam16b]
- Crestleaf (ING) - Plataforma com capacidade para armazenar as memórias familiares de forma cronológica; [Cre16]
- Find my past (ING) - Mais uma plataforma livre com possibilidade de receber sugestões de possíveis familiares; [mp16]
- Family Echo (ING) - Possibilita a criação de eventos como aniversários e a associação de fotos a esses mesmos eventos. [Ech16]

2.4 GEDCOM

GEDCOM é um formato eletrónico específico de armazenamento de dados genealógicos, permitindo assim o cruzamento de dados entre diferentes sistemas genealógicos. GEDCOM é um acrónimo proveniente de GENEalogical Data COMMunication, o tipo de arquivo pelo qual é identificado é ".ged".

O conteúdo do ficheiro é texto simples, com registos para cada indivíduo da árvore genealógica e metadados para possibilitar ligações entre eles. A maioria dos sistemas de genealogia possibilitam exportar ou importar deste formato, podendo também transformar GEDCOM em HTML recorrendo a conversores próprios para o efeito.

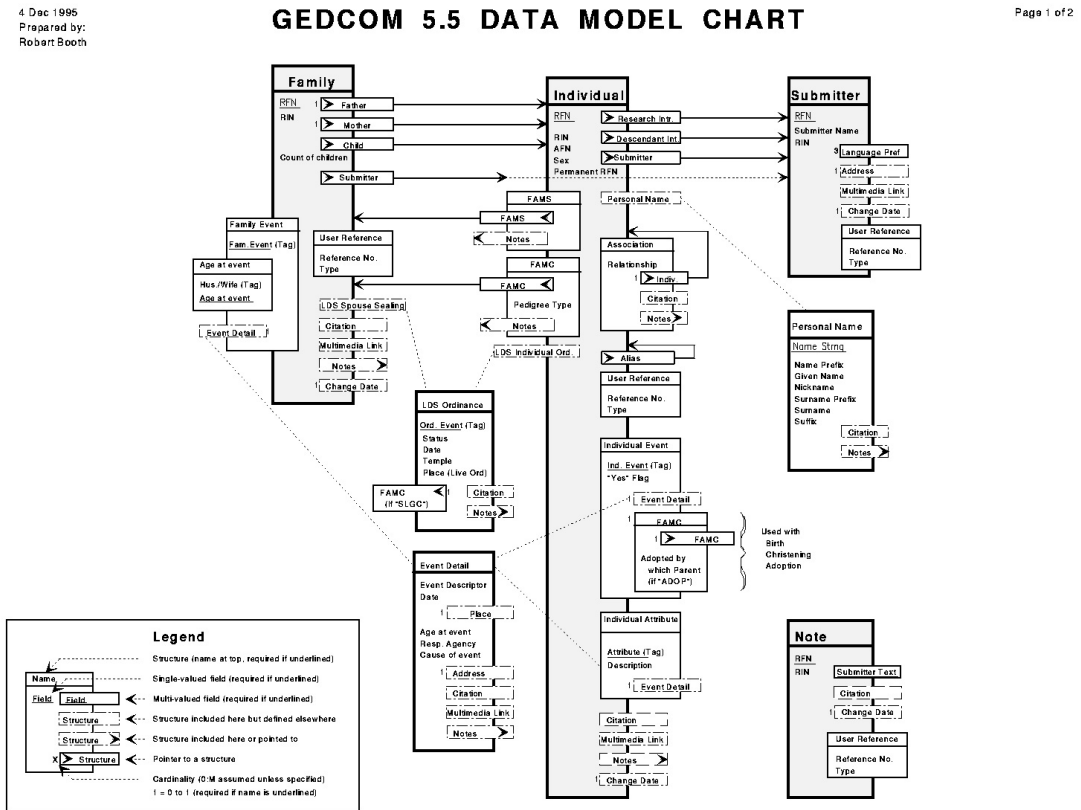


Figura 6: Modelo de Dados - parte 1

GEDCOM 5.5 DATA MODEL CHART

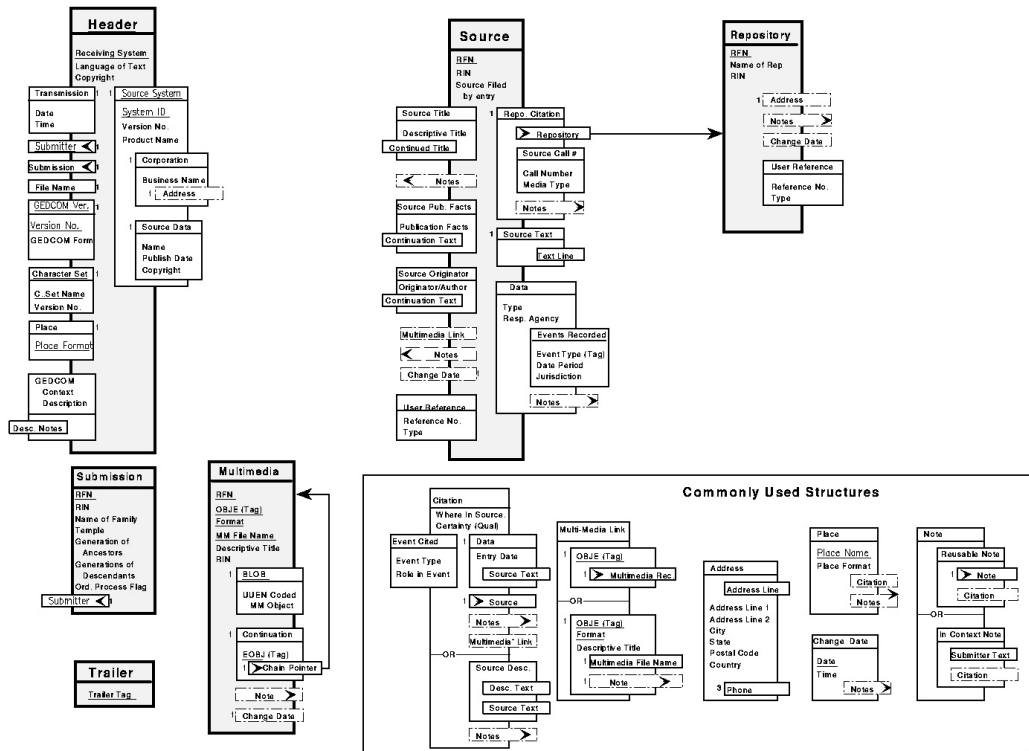


Figura 7: Modelo de Dados - parte 2

Nas duas imagens anteriores podemos ver o modelo de dados do GEDCOM [GED96], esta e outra documentação relacionada com este formato pode ser encontrada na página do GEDCOM.

## 2.5 GRAPHDB

As redes complexas são redes de dados onde a relação entre os elementos é importante tal como os próprios elementos. Através de grafos podemos fazer a sua representação sendo que a relação é descrita pelas arestas e os elementos pelos vértices. Existem alguns exemplos destas redes, tais como, sociais, de informação, tecnológicas e biológicas. [Mac11]

Este tipo de modelo de base de dados tem várias vantagens, entre elas:

- Permite uma modelação mais natural dos dados no contexto das redes complexas;
- Possibilidade de consultas que aproveitam a estrutura dos grafos;
- Trabalhando diretamente com grafos e operações relacionadas, o nível de abstracção aumenta facilitando a manipulação dos dados.

GraphDB [Ont16] é uma base de dados de grafos, compatível com os padrões W3C. Bases de dados de grafos fornecem a infra-estrutura principal para soluções onde a agilidade de modelação, integração de dados, exploração de relacionamento e publicação e consumo de dados se tornam importantes.

No GraphDB pode-se realizar inferência semântica em escala permitindo que os utilizadores criem novos fatos semânticos a partir de fatos existentes. Tem capacidade para lidar com cargas maciças, consultas e inferência em tempo real.

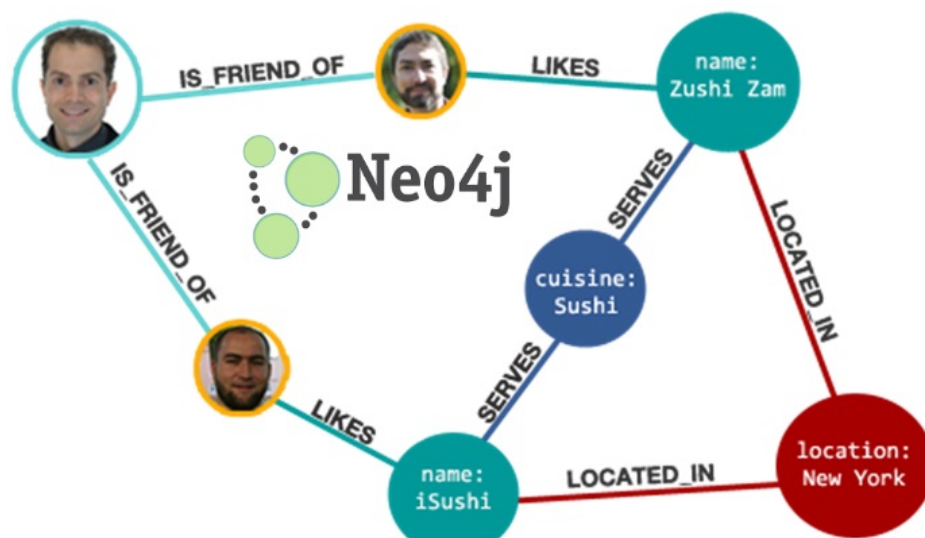


Figura 8: Exemplo de um grafo de uma rede social

## 2.6 SPARQL

SPARQL [SPA13] é simultaneamente um protocolo e uma linguagem de consulta de dados representados em RDF. Como linguagem de consulta, o SPARQL é usado para realizar queries a dados representados em RDF da mesma forma que se faz no SQL – Structured Query Language para consultar dados em bases de dados relacionais.

A sintaxe da linguagem é bastante semelhante à sintaxe SQL e disponibiliza um conjunto de comandos como SELECT, ASK, DESCRIBE e CONSTRUCT que permitem retornar dados mas não atualizá-los. Para isso é necessário utilizar uma extensão à linguagem denominada SPARUL (SPARQL UPDATE), permitindo assim efetuar comandos como INSERT, MODIFY ou DELETE.

Como protocolo, o SPARQL assenta em serviços HTTP para transmitir queries SPARQL e os resultados entre uma aplicação cliente e um SPARQL Endpoint. Estes terminais são aplicações Web, que oferecem um interface para pedidos do tipo HTTP, GET ou POST, para acesso a um conjunto de dados RDF. Neste contexto, denomina-se de SPARQL Endpoint um servidor HTTP que expõe dados através do protocolo SPARQL. [Roc15b]

---

## MODELAÇÃO E CONSTRUÇÃO DA ONTOLOGIA

---

### 3.1 DESENVOLVIMENTO DA ONTOLOGIA

A metodologia usada neste desenvolvimento foi a Metodologia 101 (Ontology Development 101). No Estado da Arte essa metodologia foi descrita de forma generalizada, neste capítulo será agora descrito o desenvolvimento específico desta ontologia tendo em conta os passos da metodologia aplicada.

O formato do OWL escolhido para a ontologia criada foi o OWL DL, dado que o OWL Lite não tem capacidade para dar resposta ao pretendido e o OWL Full não garante a completitude e decidibilidade computacional necessária.

Para desenvolver a ontologia foi usado o Protégé, o Protégé é um editor genérico de ontologias e bases de conhecimento, existem outras ferramentas do mesmo género, tais como: Ontoedit, OilEd entre outros. Usando o Protégé foi possível criar a ontologia através de uma interface gráfica, sendo produzida no fim uma representação OWL da mesma.

#### 3.1.1 *Determinar Contexto*

O tema da ontologia é a genealogia, dentro deste tema abrangente a prioridade foram as relações de parentesco entre os indivíduos de uma família, eventos familiares e documentos que podem ser associados a um evento ou a um indivíduo. Este é o contexto da ontologia.

### 3.1.2 Considerar Reutilização

Uma etapa importante é averiguar se há ontologias já desenvolvidas que possam ser usadas, isto pode poupar tempo e problemas posteriormente na especificação da ontologia.

Existem alguns trabalhos na área da genealogia disponíveis na Internet mas nenhum dos que foi possível estudar tinha acessível uma ontologia a partir da qual fosse possível trabalhar.

A única reutilização que foi feita, neste caso, foi usar uma ontologia que tinha sido elaborada no decorrer de uma Unidade Curricular do Mestrado no ano letivo anterior, mas que ainda estava num estado muito inicial de desenvolvimento.

### 3.1.3 Enumerar Conceitos

Com o contexto definido deve-se enumerar os conceitos que se vão ter em consideração, surgindo assim possíveis classes, propriedades ou até elementos.

Pessoa, Evento e Documento são os principais conceitos. Tendo em conta o carácter cronológico a data é também um conceito importante e claro as relações de parentesco.

### 3.1.4 Definir Classes

Para definir as classes optou-se por uma hierarquia Top-Down com as três classes principais a serem: "Person", "Event" e "Document" representativas do contexto identificado anteriormente.

Além destas, existem mais duas classes, a "Family" e a "FamilyRole" fazendo com que a ontologia esteja também preparada para identificar sub grupos na árvore familiar que serão uma espécie de agregados familiares compostos pelo pai, mãe e respetivos filhos.

De uma forma sucinta, uma "Family" pode ter vários "FamilyRole" e representa um agregado familiar, já o "FamilyRole" funciona como um papel que a pessoa tem dentro de um desses agregados, o "FamilyRole" tem 3 "role" possíveis que são: "Dad", "Mom" e "Son", sendo que uma instância do "FamilyRole" é constituída pelo papel e a pessoa a que se associa. Já a instância da Classe "Family" pode conter um ou mais "FamilyRole".

Código exemplo de uma classe definida na Ontologia:

```
<Declaration>
  <Class IRI="#Person" />
</Declaration>
```

No diagrama abaixo podem-se ver as classes da ontologia e a ligação entre elas caso exista essa ligação.

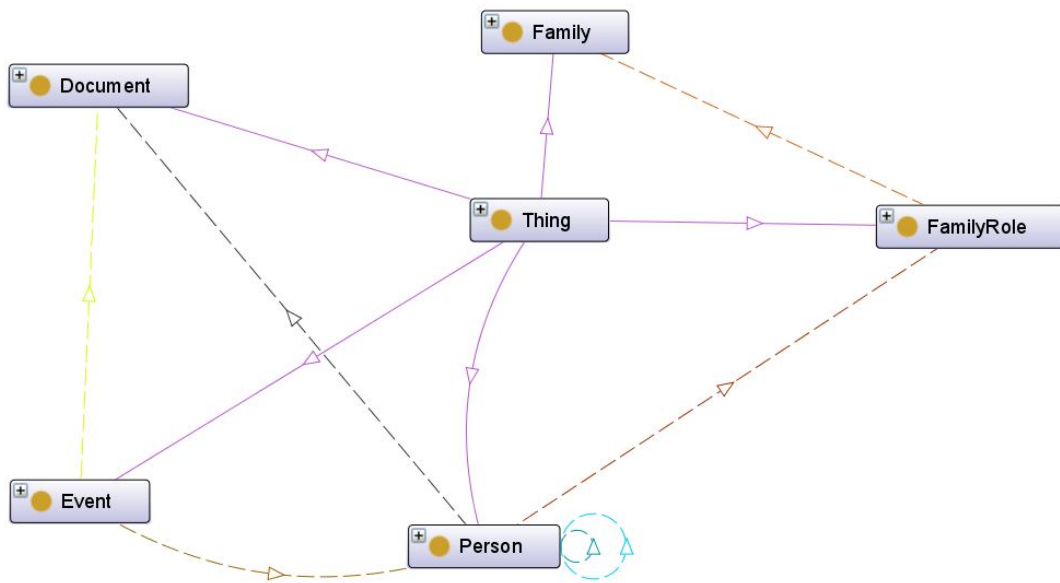


Figura 9: Classes da Ontologia

Já no esquema da página seguinte pode ser visualizada a "árvore" de classes e subclasses da ontologia que são tidas em conta para a aplicação.



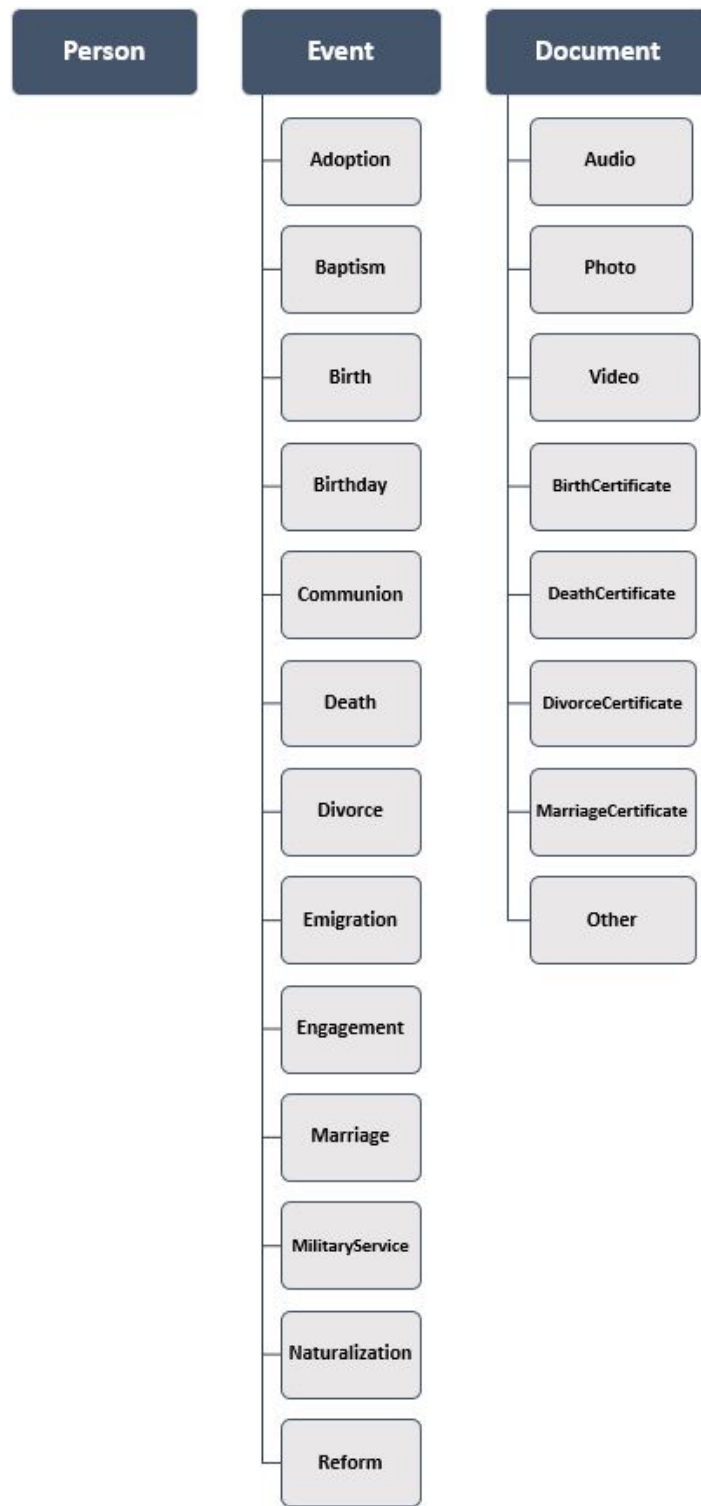


Figura 10: Classes e Subclasses da Ontologia - Aplicação Web

### 3.1.5 Definir Propriedades

Nos diagramas seguintes estão enumeradas todas as propriedades das instâncias que foram consideradas oportunas para a construção da ontologia, associadas respetivamente a cada classe ou subclasse a que estão restritas.

Código exemplo de uma propriedade definida na Ontologia:

```
<Declaration>
  <DataProperty IRI="#address"/>
</Declaration>
```

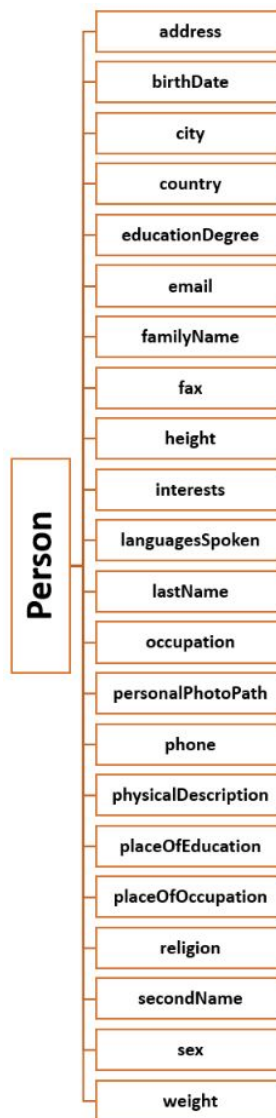


Figura 11: Diagrama de Propriedades da Classe - Person

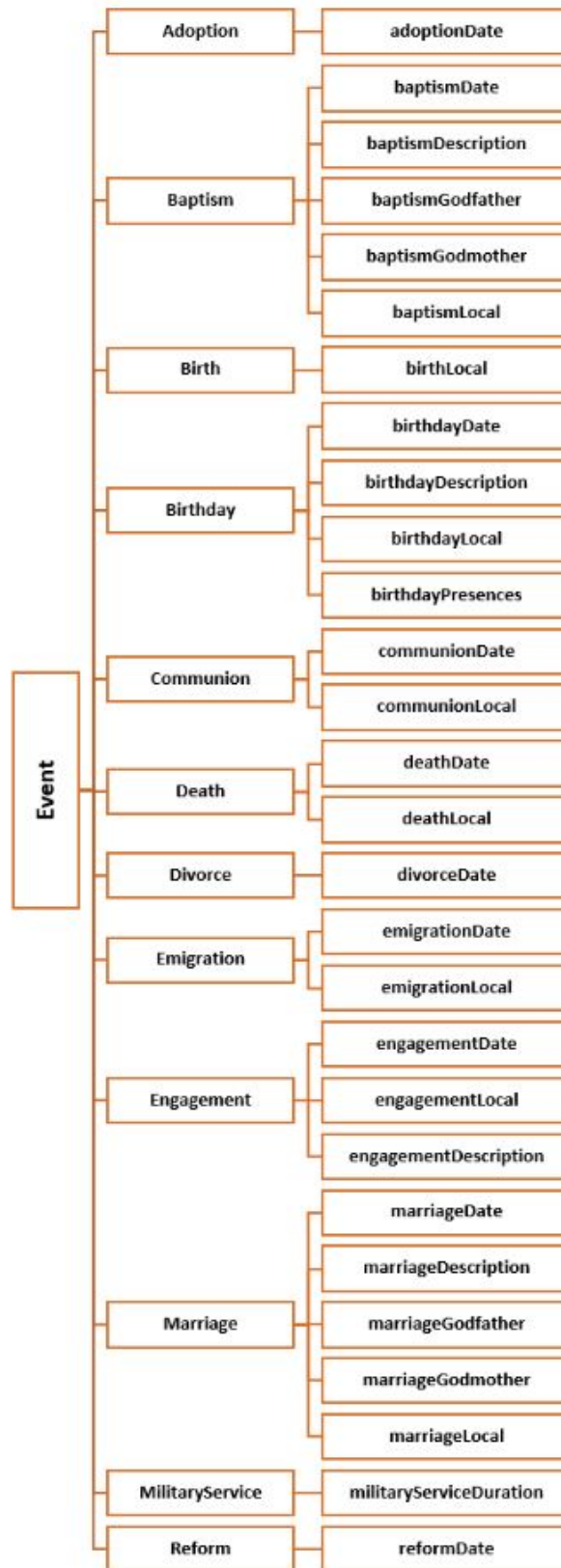


Figura 12: Diagrama de Propriedades da Classe - Event

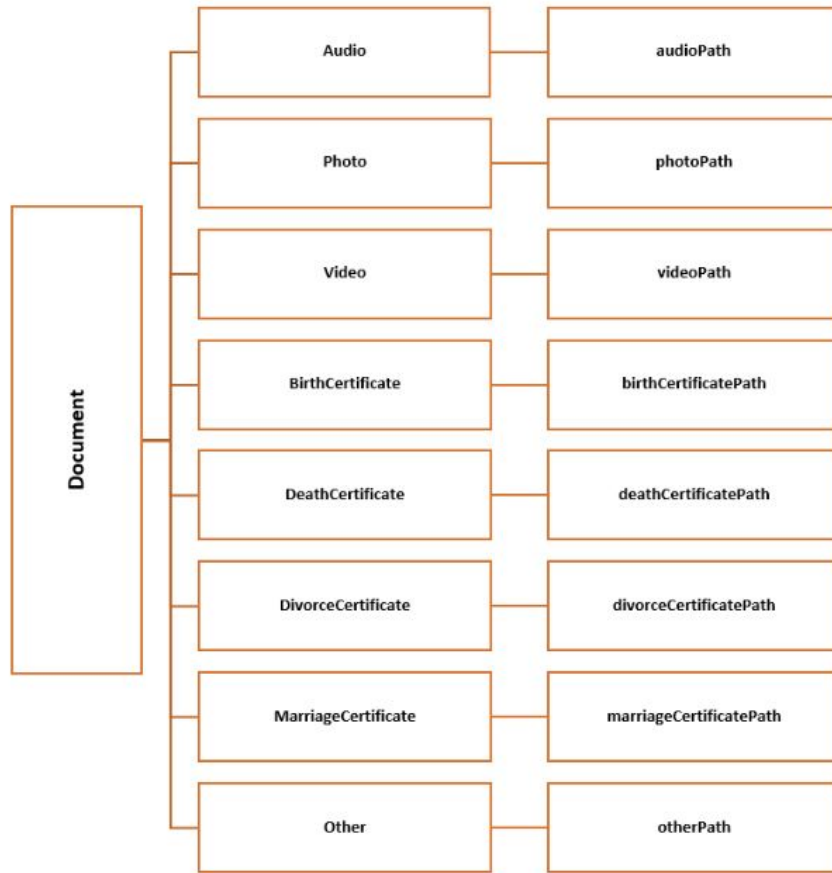


Figura 13: Diagrama de Propriedades da Classe - Document

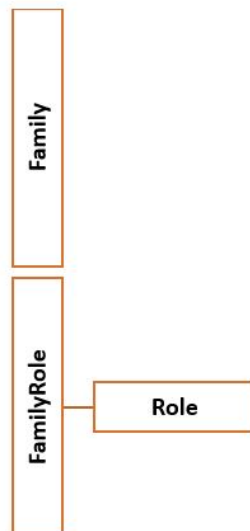


Figura 14: Diagrama de Propriedades das Classes - Family e FamilyRole

### Relações

Tratando-se de uma parte importante para tornar possíveis as inferências semânticas, as relações criadas serão apresentadas de forma extensiva no formato de tabela com o nome da relação e todas as características associadas à mesma.

Existem quatro relações nestas tabelas que não serão usadas na aplicação web, mas que foram criadas na ontologia de forma a precaver no futuro uma possível utilização das mesmas nos relacionamentos entre as classes "Family" e "FamilyRole" para os efeitos mencionados na secção (definir classes).

As quatro relações mencionadas são: "familyHas", "familyRolePerson", "hasFamilyRole" e "inFamily".

Estando o editor em inglês e as relações definidas também nessa língua não será feita tradução de forma a não perder significado com a mesma. A tabela tem sete colunas sendo que:

- Object Properties - É o nome da relação;
- SubProperty Of - É uma sub relação de outra relação;
- Inverse Of - É a relação inversa da mesma;
- Domain - Identifica a classe dos indivíduos que representam o domínio;
- Range - Identifica a classe dos indivíduos que representam a imagem;
- SuperProperty Of (Chain) - Inferência a ser feita a partir da existência de outra relação, ou seja, se A ou B forem verdadeiros C aplica-se;
- Characteristics - Propriedades que podem ser associadas a uma relação, existem 8 (functional, inverse functional, transitive, symmetric, asymmetric, reflexive, irreflexive);

Código exemplo de uma relação definida na Ontologia:

```
<Declaration>
  <ObjectProperty IRI="#hasRelationship"/>
</Declaration>
```

<u>Object Properties</u>	<u>SubProperty Of</u>	<u>Inverse Of</u>	<u>Domain</u>	<u>Range</u>	<u>SuperProperty Of (Chain)</u>	<u>Characteristics</u>
<u>docHasEvent</u>		<u>eventHasDoc</u>				
<u>docHasPerson</u>		<u>personHasDoc</u>				
<u>eventHasDoc</u>		<u>docHasEvent</u>	<u>Event</u>	<u>Document</u>		
<u>eventHasPerson</u>		<u>personHasEvent</u>	<u>Event</u>	<u>Person</u>		
<u>familyHas</u>		<u>inFamily</u>				
<u>familyRolePerson</u>		<u>hasFamilyRole</u>				
<u>hasFamilyRole</u>		<u>familyRolePerson</u>	<u>Person</u>	<u>FamilyRole</u>		
<u>hasRelationship</u>		<u>hasRelationship</u>	<u>Person</u>	<u>Person</u>		<u>Symmetric</u>
<u>hasAncestor</u>	<u>Inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>isAncestorOf</u>				<u>Transitive</u>
<u>hasGrandParent</u>	<u>hasAncestor inverse</u> <u>(isAncestorOf)</u>	<u>isGrandParentOf</u>			<u>hasProgenitor o</u> <u>hasProgenitor</u> <u>SubPropertyOf</u> <u>hasGrandParent</u>	
<u>hasGrandFather</u>	<u>hasGrandParent</u> <u>inverse(isGrandParentOf)</u>	<u>isGrandFatherOf</u>	<u>Person</u>	<u>Man</u>	<u>hasProgenitor o hasDad</u> <u>SubPropertyOf</u> <u>hasGrandFather</u>	
<u>hasGrandMother</u>	<u>inverse (isGrandParentOf)</u> <u>hasGrandParent</u>	<u>isGrandMotherOf</u>	<u>Person</u>	<u>Woman</u>	<u>hasProgenitor o</u> <u>hasMom SubPropertyOf</u> <u>hasGrandMother</u>	
<u>hasProgenitor</u>	<u>inverse (isAncestorOf)</u> <u>has Ancestor</u>	<u>isProgenitorOf</u>				
<u>hasDad</u>	<u>hasProgenitor</u> <u>inverse (isProgenitorOf)</u>	<u>isDadOf</u>	<u>Person</u>	<u>Man</u>		<u>Functional</u>
<u>hasMom</u>	<u>hasProgenitor</u> <u>inverse (isProgenitorOf)</u>	<u>isMomOf</u>	<u>Person</u>	<u>Woman</u>		<u>Functional</u>

Figura 15: Tabela de Relações - Parte 1

<u>Object Properties</u>	<u>SubProperty Of</u>	<u>Inverse Of</u>	<u>Domain</u>	<u>Range</u>	<u>SuperProperty Of (Chain)</u>	<u>Characteristics</u>
<u>hasParentsInLaw</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>					
<u>hasFatherInLaw</u>	<u>hasParentInLaw</u> <u>inverse (isParentsInLawOf)</u>	<u>isFatherInLawOf</u>	Person	Man	<u>isSpouseOf</u> o <u>hasDad</u> <u>SubPropertyOf</u> <u>hasFatherInLaw</u>	
<u>hasMotherInLaw</u>	<u>hasParentInLaw</u> <u>inverse (isParentsInLawOf)</u>	<u>isMotherInLawOf</u>	Person	Woman	<u>isSpouseOf</u> o <u>hasMom</u> <u>SubPropertyOf</u> <u>hasMotherInLaw</u>	
<u>hasSister_BrotherInLaw</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>isSister_BrotherInLawOf</u>	Person			
<u>hasUncle_Aunt</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>isUncle_AuntOf</u>	Person			
<u>hasUncle_AuntInLaw</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>isUncle_AuntInLawOf</u>	Person			
<u>isAncestorOf</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>has Ancestor</u>				Transitive
<u>isGrandParentOf</u>	<u>inverse (has Ancestor)</u> <u>isAncestorOf</u>	<u>hasGrandParent</u>				
<u>isGrandFatherOf</u>	<u>inverse (hasGrandParent)</u> <u>isGrandParentOf</u>	<u>hasGrandFather</u>				
<u>isGrandMotherOf</u>	<u>isGrandParentOf</u> <u>inverse (hasGrandParent)</u>	<u>hasGrandMother</u>				
<u>isProgenitorOf</u>	<u>inverse (has Ancestor)</u> <u>isAncestorOf</u>	<u>hasProgenitor</u>				
<u>isDadOf</u>	<u>inverse (hasProgenitor)</u> <u>isProgenitorOf</u>	<u>hasDad</u>				Inverse Functional
<u>isMomOf</u>	<u>inverse (hasProgenitor)</u> <u>isProgenitorOf</u>	<u>hasMom</u>				Inverse Functional
<u>isCousinOf</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>isCousinOf</u>	Person		<u>hasProgenitor</u> o <u>isSiblingOf</u> o <u>isProgenitorOf</u> <u>SubPropertyOf</u> <u>isCousinOf</u>	Symmetric

Figura 16: Tabela de Relações - Parte 2

<u>Object Properties</u>	<u>SubProperty Of</u>	<u>Inverse Of</u>	<u>Domain</u>	<u>Range</u>	<u>SuperProperty Of (Chain)</u>	<u>Characteristics</u>
<u>isParentInLawOf</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>					
<u>isFatherInLawOf</u>	<u>inverse (hasParentInLaw)</u> <u>isParentInLawOf</u>	<u>hasFatherInLaw</u>				
<u>isMotherInLawOf</u>	<u>inverse (hasParentInLaw)</u>	<u>hasMotherInLaw</u>				
<u>isSiblingOf</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>isSiblingOf</u>	Person			Symmetric
<u>isSister_BrotherInLawOf</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>hasSister_BrotherInLaw</u>			<u>isSiblingOf o isSpouseOf</u> <u>SubPropertyOf</u> <u>isSister_BrotherInLawOf</u>	
<u>isSpouseOf</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>isSpouseOf</u>	Person	Person		Symmetric
<u>isUncle_AuntInLawOf</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>hasUncle_AuntInLaw</u>			<u>isSpouseOf o isSiblingOf</u> <u>o isProgenitorOf</u> <u>SubPropertyOf</u> <u>isUncle_AuntInLawOf</u>	
<u>isUncle_AuntOf</u>	<u>inverse (hasRelationship)</u> <u>hasRelationship</u>	<u>hasUncle_Aunt</u>			<u>isSiblingOf o</u> <u>isProgenitorOf</u> <u>SubPropertyOf</u> <u>isUncle_AuntOf</u>	
<u>inFamily</u>		<u>familyHas</u>	<u>FamilyRole</u>	Family		
<u>personHasDoc</u>		<u>docHasPerson</u>	Person	Document		
<u>personHasDoc</u>		<u>eventHasPerson</u>	Person	Event		

Figura 17: Tabela de Relações - Parte 3



## Inferência

Através da definição de axiomas na forma de cadeia de propriedades é possível realizar inferências. O recurso a estes axiomas possibilita que muitas das relações de parentesco sejam inferidas sem ser necessária a intervenção do utilizador.

### Inferência para Avós (Inverse Of)

Na relação "hasGrandFather" as inferências podem surgir do "Inverse Of" que basicamente funciona do seguinte modo:

- Premissa - Daniel é neto do Pedro.
- Inferência - Pedro é avô do Daniel.

Ou seja, define uma relação inversa à relação existente.

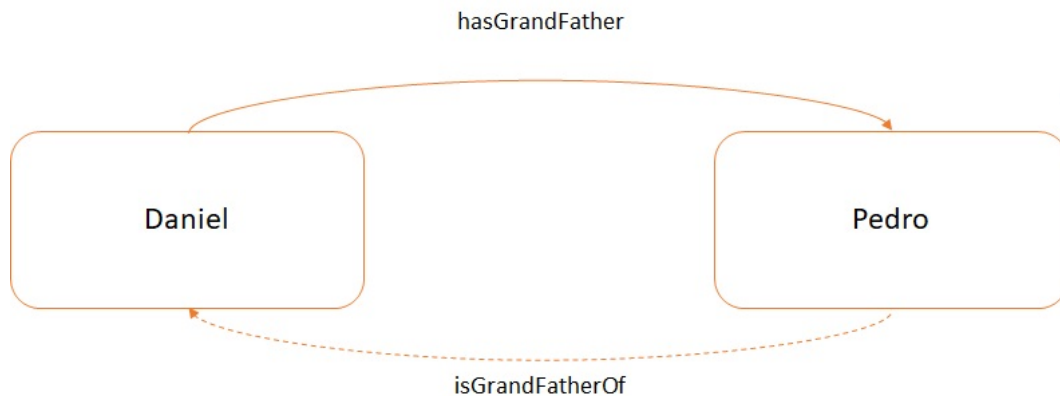


Figura 18: Esquema "Inverse Of" da inferência para Avós

Código da Inferência definida na Ontologia:

```
<InverseObjectProperties>
  <ObjectProperty IRI="#hasGrandParent"/>
  <ObjectProperty IRI="#isGrandParentOf"/>
</InverseObjectProperties>
```

### Inferência para Avós (SuperProperty Of (Chain))

Existem contudo inferências mais complexas que as "Inverse Of", nas tabelas de relações essas são as que estão na coluna "SuperProperty Of (Chain)". Usando a mesma relação do ponto anterior, "hasGrandFather" a inferência agora é realizada da seguinte forma:

- Premissa 1 - Daniel tem como pai o Luís.
- Premissa 2 - Luís tem como pai o Pedro.
- Inferência - O Daniel tem como avô o Pedro.

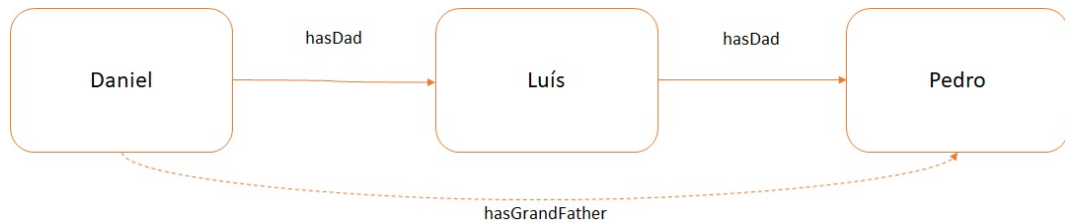


Figura 19: Esquema "SuperProperty Of (Chain)" da inferência para Avós

Código da Inferência definida na Ontologia:

```
<SubObjectPropertyOf >
  <ObjectPropertyChain >
    <ObjectProperty IRI="#hasProgenitor" />
    <ObjectProperty IRI="#hasDad" />
  </ObjectPropertyChain >
  <ObjectProperty IRI="#hasGrandFather" />
</SubObjectPropertyOf >
```

### Inferência para Tios "de sangue"(SuperProperty Of (Chain))

Para se encontrar o parentesco de tios/tias, o axioma usado é o seguinte:

- Premissa 1 - O Tiago é irmão do André.
- Premissa 2 - André é pai do João.
- Inferência - Tiago é tio do João.

De uma forma sucinta procura-se por irmãos da pessoa A, se o irmão ou irmã dessa pessoa A for progenitor de alguém (pessoa B), assume-se que a pessoa A é tio ou tia da pessoa B.

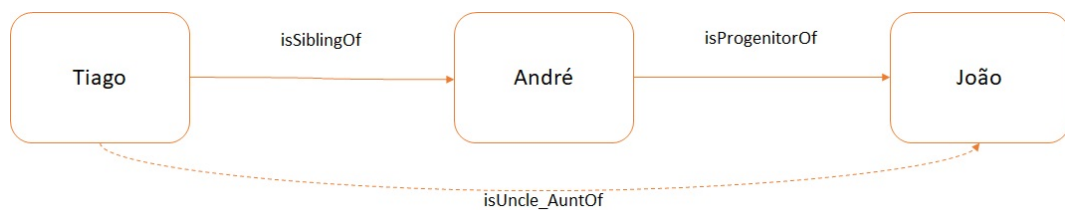


Figura 20: Esquema "SuperProperty Of (Chain)" da inferência para Tios

Código da Inferência definida na Ontologia:

```
SubObjectPropertyOf >
  <ObjectPropertyChain >
    <ObjectProperty IRI="#isSiblingOf"/>
    <ObjectProperty IRI="#isProgenitorOf"/>
  </ObjectPropertyChain >
  <ObjectProperty IRI="#isUncle_AuntOf"/>
</SubObjectPropertyOf >
```

### Inferência para Tios "de casamento"(SuperProperty Of (Chain))

Outra relação em que é necessário um axioma é a dos tios por casamento, sendo que neste caso a cadeia de propriedade é constituída por três premissas:

- Premissa 1 - O Alfredo é casado com a Tatiana.
- Premissa 2 - A Tatiana é irmã da Alice.
- Premissa 3 - A Alice é mãe do Cristiano.
- Inferência - Alfredo é tio por casamento do Cristiano.

O que se pretende inferir neste caso são aquelas pessoas que passam a ser tios quando se casam com um irmão dos pais de um indivíduo.

Código da Inferência definida na Ontologia:

```
<SubObjectPropertyOf >
  <ObjectPropertyChain >
    <ObjectProperty IRI="#isSpouseOf"/>
    <ObjectProperty IRI="#isSiblingOf"/>
    <ObjectProperty IRI="#isProgenitorOf"/>
  </ObjectPropertyChain >
  <ObjectProperty IRI="#isUncle_AuntInLawOf"/>
</SubObjectPropertyOf >
```

### Inferência para Sogros (SuperProperty Of (Chain))

Para alguém ter sogros é preciso que seja casado com outra pessoa e que essa pessoa tenha pais. De modo a obter esta cadeia de propriedades o axioma baseia-se no seguinte:

- Premissa 1 - O José é casado com a Ana.
- Premissa 2 - A Ana é filha do António.

- Inferência - O José tem como sogro o António.

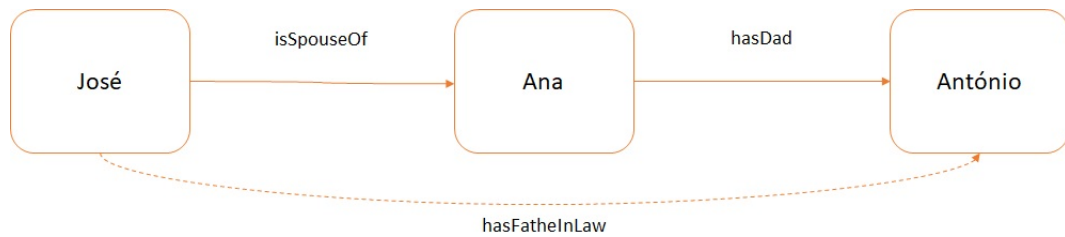


Figura 21: Esquema "SuperProperty Of (Chain)" da inferência para Sogros

Código da Inferência definida na Ontologia:

```
<SubObjectPropertyOf >
  <ObjectPropertyChain >
    <ObjectProperty IRI="#isSpouseOf" />
    <ObjectProperty IRI="#hasDad" />
  </ObjectPropertyChain >
  <ObjectProperty IRI="#hasFatherInLaw" />
</SubObjectPropertyOf >
```

### Inferência para Primos (SuperProperty Of (Chain))

O axioma que infere a relação de primos tem de passar por três validações antes de ser definido:

- Premissa 1 - O Rui é filho do Miguel.
- Premissa 2 - O Miguel é irmão da Maria.
- Premissa 3 - A Maria é mãe do Rafael.
- Inferência - O Rui é primo do Rafael.

Resumidamente, a pessoa A tem de ser filho de alguém, depois o seu pai ou mãe tem de ter um irmão ou irmã que por sua vez deve ser pai ou mãe da pessoa B, fazendo com que a pessoa A seja primo/a da pessoa B.

Código da Inferência definida na Ontologia:

```
<SubObjectPropertyOf >
  <ObjectPropertyChain >
    <ObjectProperty IRI="#hasProgenitor" />
    <ObjectProperty IRI="#isSiblingOf" />
    <ObjectProperty IRI="#isProgenitorOf" />
  </ObjectPropertyChain >
</SubObjectPropertyOf >
```

```

</ObjectPropertyChain>
<ObjectProperty IRI="#isCousinOf"/>
</SubObjectPropertyOf>

```

### Inferência para Cunhados (SuperProperty Of (Chain))

Cunhados são os irmãos da pessoa com quem se contraiu um matrimónio. Para se obter esta relação o axioma procura verificar se a pessoa é casada e se o esposo/esposa tem irmãos.

- Premissa 1 - O Alexandre é casado com a Isabel.
- Premissa 2 - A Isabel é irmã da Cátia.
- Inferência - A Cátia é cunhada do Alexandre.

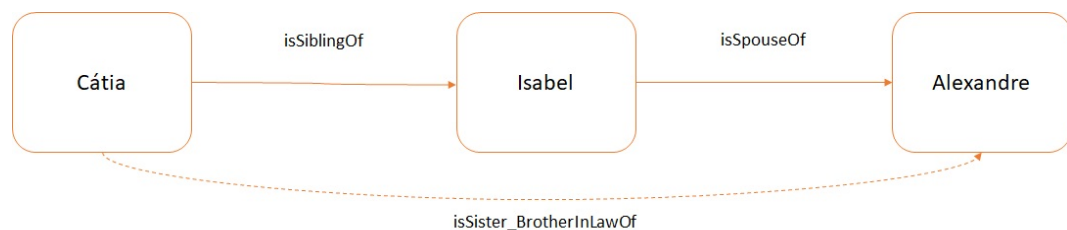


Figura 22: Esquema "SuperProperty Of (Chain)" da inferência para Cunhados

Código da Inferência definida na Ontologia:

```

<SubObjectPropertyOf>
  <ObjectPropertyChain>
    <ObjectProperty IRI="#isSiblingOf"/>
    <ObjectProperty IRI="#isSpouseOf"/>
  </ObjectPropertyChain>
  <ObjectProperty IRI="#isSister_BrotherInLawOf"/>
</SubObjectPropertyOf>

```

### Irmãos

Das relações principais de parentesco apenas a do irmão não foi possível inferir já que ao definir essa relação como "InverseOf" a inferência era redundante o que gerava por exemplo que o Daniel fosse irmão do Ricardo mas também fosse irmão dele próprio.

Aparentemente este problema seria resolvido usando a propriedade reflexiva para definir esta relação mas isso também não funcionou já que foram testados vários motores para

executar a ontologia e eles não aguentam com cadeias de propriedades em que haja uma propriedade reflexiva, acabando invariavelmente por rebentar com a ontologia. Este é um problema já detetado nos motores de inferência, não havendo ainda uma solução para o mesmo.

Na aplicação, como será visto posteriormente, é possível forçar essa relação, que mesmo não sendo de todo o cenário ideal, foi a solução possível para se contornar este problema.

### 3.1.6 Criar Instâncias

As instâncias são criadas pelo utilizador que irá inserir os elementos da sua família, os eventos e os documentos que pense serem úteis para a sua história familiar.

Contudo, já foram inseridas algumas instâncias na ontologia para se poder testar a mesma, o exemplo usado foi de uma pequena família não sendo os dados totalmente verídicos.

Código exemplo de uma instância definida na Ontologia:

```
<Declaration>  
  <NamedIndividual IRI="#Frederico_Moreira_Ribeiro"/>  
</Declaration>
```

---

## IMPLEMENTAÇÃO

---

### 4.1 ARQUITETURA GERAL

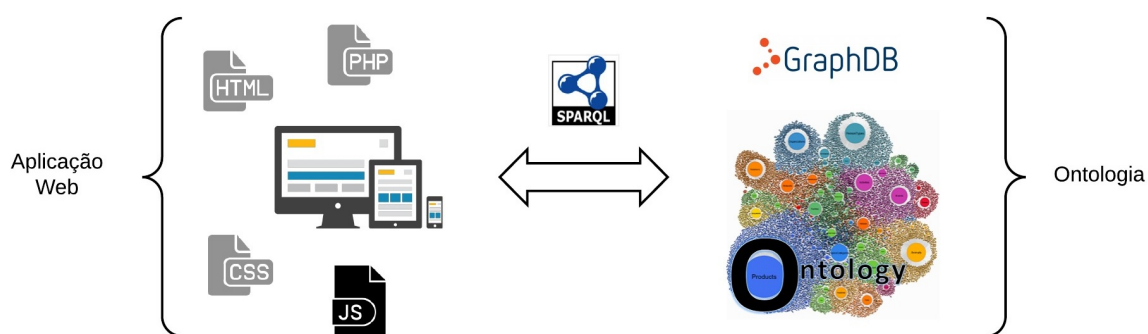


Figura 23: Arquitetura

Como se vê no esquema anterior, é fácil perceber que esta dissertação consiste num sistema com duas partes distintas mas que comunicam entre si. A primeira parte é a da ontologia e respectivo repositório e a segunda parte consiste na aplicação web que irá alimentar e ser alimentada pela ontologia.

No esquema já estão representadas as tecnologias e linguagens usadas, fazendo com que se tenha uma visão geral desde já, sendo que ao longo da dissertação irão ser abordadas de forma mais detalhada todas essas particularidades.

## 4.2 REPOSITÓRIO PARA A ONTOLOGIA

A ontologia após o que foi exposto no capítulo de modelação está pronta a ser colocada numa base de dados de grafos, que como foi referido antes, será o GraphDB.

De modo a ser possível alimentar a aplicação web tornou-se necessário ter uma base de dados de grafos onde se colocou a ontologia criada de forma a realizar-se as inferências semânticas necessárias, tirando assim todo o proveito da mesma.

Para tal efeito foi usado o GraphDB, existem outras opções no que toca a base de dados de grafos mas após avaliação de algumas dessas hipóteses optou-se pelo GraphDB por se adaptar ao que é pretendido.

### 4.2.1 Importar Ontologia no GraphDB

O primeiro passo foi o de instalar o GraphDB Free. Após instalar e definir a porta que será usada podemos abrir o GraphDB que automaticamente abre no browser um workbench para fazermos a gestão dos repositórios e dos grafos que lá existam.

Antes de importar no GraphDB, temos de exportar, com recurso ao Protégé, um ficheiro da ontologia em formato RDF. Relembrar que a ontologia foi desenvolvida em OWL mas como o GraphDB apenas importa em formato RDF é preciso exportar a ontologia nesse formato. Refira-se que as vantagens inerentes à especificação em OWL não são perdidas com este processo caso seja feito o aconselhado no parágrafo seguinte.

No GraphDB cria-se um novo repositório com o nome desejado, além disso aconselha-se que se altere o Ruleset para "OWL2 otimizado" para ser possível fazer as inferências definidas dos dados adicionados posteriormente à ontologia.



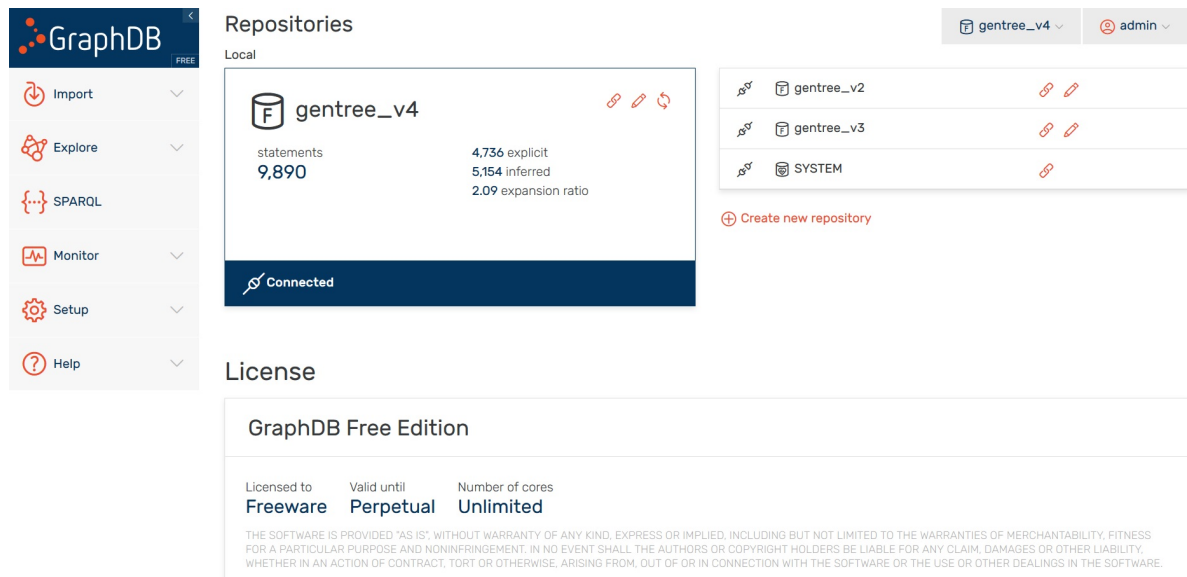


Figura 24: Interface do GraphDB - Repositórios

Nesta fase basta apenas importar o RDF para o GraphDB e a ontologia fica pronta para ser explorada.

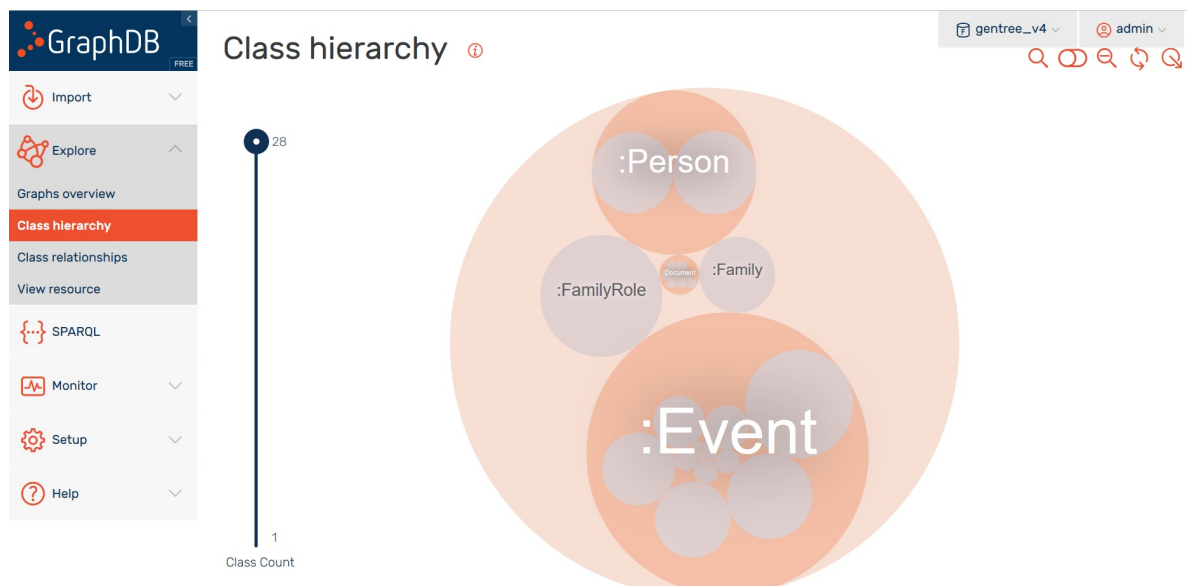


Figura 25: Interface do GraphDB - Classes

### 4.2.2 Utilidades do GraphDB

Além do já mencionado, o GraphDB possui também um separador no workbench onde se pode fazer queries ao grafo, esta ferramenta mostrou-se muito útil para a construção das queries usadas na comunicação com a aplicação web.

Ora esta ferramenta possibilitou que antes de incorporar alguma query nas páginas da aplicação fosse possível testar primeiro diretamente no GraphDB, visualizando assim os resultados obtidos da query e possíveis erros que a mesma podia ter.

The screenshot shows the GraphDB SPARQL Query & Update interface. The left sidebar contains navigation options: Import, Explore, SPARQL (selected), Monitor, Setup, and Help. The main area displays a SPARQL query with the following content:

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX owl: <http://www.w3.org/2002/07/owl#>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
6
7 SELECT ?person
8 WHERE {
9   ?person a gentree:Person.
10
11 } LIMIT 100

```

Below the query editor, there are tabs for Table, Raw Response, Pivot Table, Google Chart, and Graph(beta). The results are displayed in a table with 5 rows of data:

	person
1	gentree:Adriana_Sofia_Gomes_Ribeiro_Moreira_Santos
2	gentree:Alberto_Ribeiro
3	gentree:Alexandra_Ribeiro_Araujo
4	gentree:André_Ribeiro_Araujo
5	gentree:António_Luis_Moreira_Santos

Figura 26: Interface do GraphDB - SPARQL

O workbench do GraphDB possui outras funções interessantes, mas a outra ferramenta que é de salientar não pertence ao workbench, é o Ontotext Cognitive Cloud que permite o acesso externo ao grafo, o que numa perspetiva de colocar o sistema aberto para todos é muito útil.

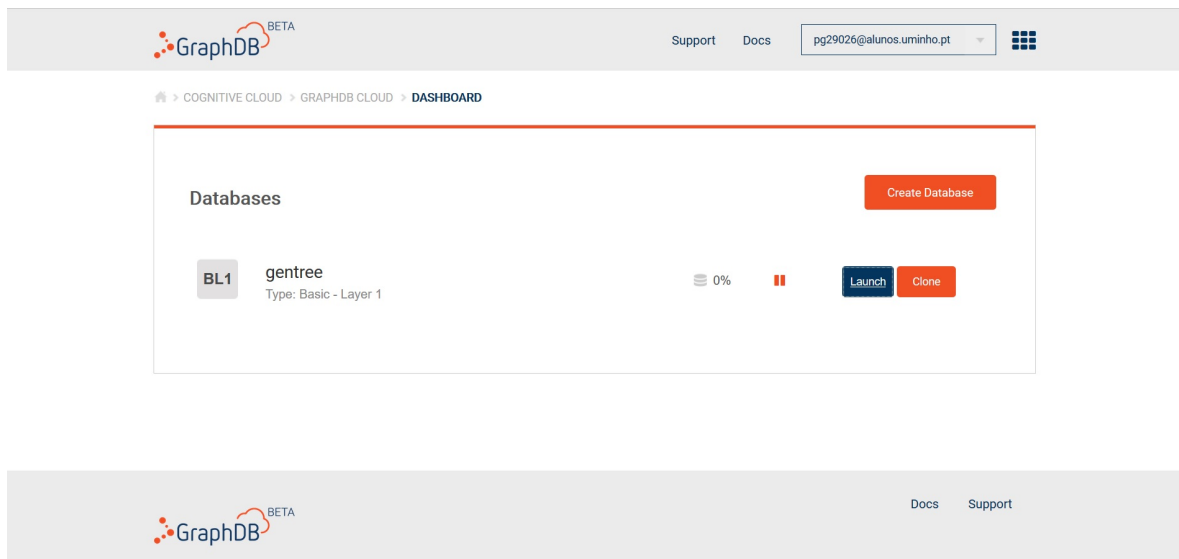


Figura 27: Interface do Ontotext Cognitive Cloud

## 4.3 XAMPP

Para correr a aplicação web é necessário um servidor, e é para este efeito que o XAMPP foi usado, já que se trata de um pacote de serviços que engloba entre outros o Apache e o PHP, servindo desta forma para testar o trabalho efetuado ao longo deste projeto.

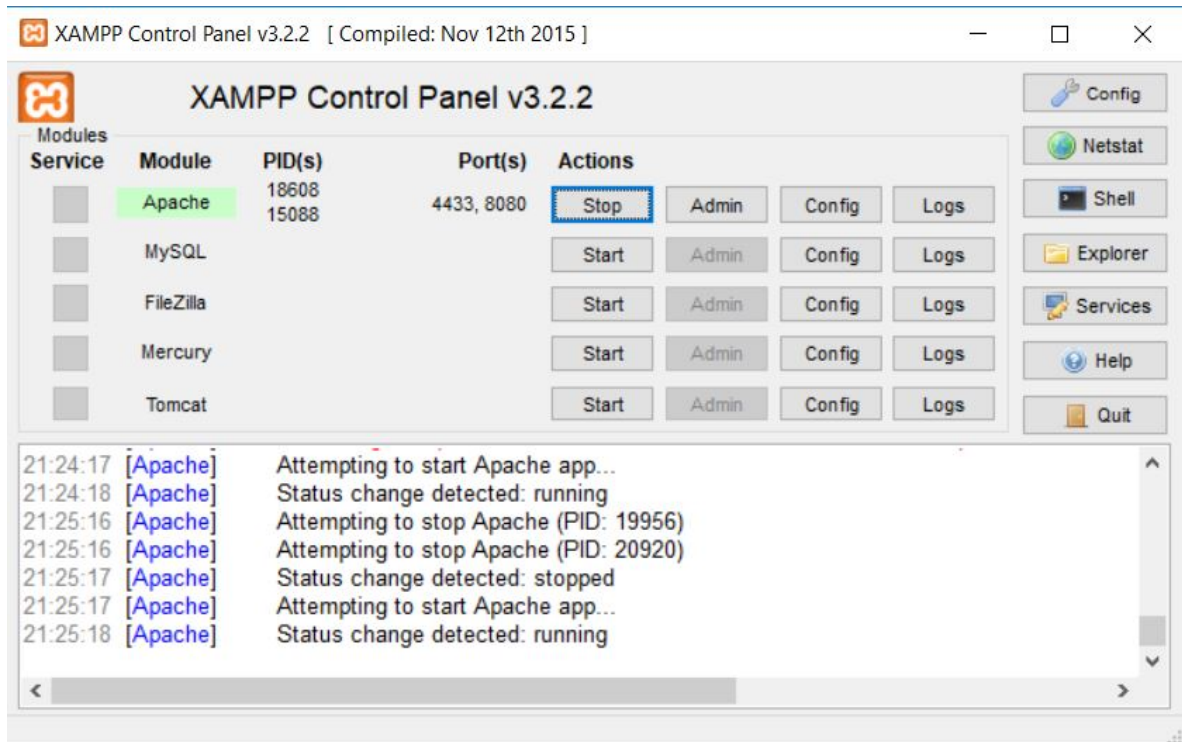


Figura 28: Interface do XAMPP

#### 4.4 COMUNICAÇÃO ENTRE A APLICAÇÃO WEB E O GRAPHDB

Identificadas estas duas partes é preciso fazê-las comunicar de forma a que a informação existente na ontologia passe para as páginas da aplicação web e no sentido contrário que as informações adicionadas e/ou alteradas sejam transmitidas para a ontologia.

Esta ponte é feita através do SPARQL, uma linguagem de consulta a dados em RDF, que também funciona como protocolo assentando em serviços HTTP para ser possível transmitir os resultados para uma aplicação cliente, que neste caso é a aplicação web que foi criada.

##### 4.4.1 *Sparqllib*

De forma a tornar este processo menos complexo, usou-se o Sparqllib que é uma biblioteca PHP simples para consultar o SPARQL a partir do PHP.

Um breve excerto do sparqllib que mostra a definição da função "get":

```
<?php
function sparql_get( $endpoint, $sparql )
{
    $db = sparql_connect( $endpoint );
    if( !$db ) { return; }
    $result = $db->query( $sparql );
    if( !$result ) { return; }
    return $result->fetch_all();
}
?>
```

## 4.5 APLICAÇÃO WEB

### 4.5.1 HTML

**HTML** [HTM17] é uma abreviação de Hypertext Markup Language e trata-se de uma linguagem de marcação usada na estruturação de páginas web. A sua sintaxe é simples e é baseada em tags tal como acontece no caso do XML, sendo que esses tags representam os vários elementos de uma página.

Quando se acede a uma página web através de um navegador, ele é capaz de interpretar o HTML e compilá-lo de forma a ser perceptível para o utilizador, exibindo textos, botões, entre outros componentes com as configurações definidas através das várias tags que o HTML possui.

Nos dias de hoje o HTML está na versão 5 e é padronizado pelo W3C (World Wide Web Consortium), uma organização internacional responsável por definir padrões para a internet, como é o caso de outras linguagens como o XML, as CSS e o SOAP. [Rod16]

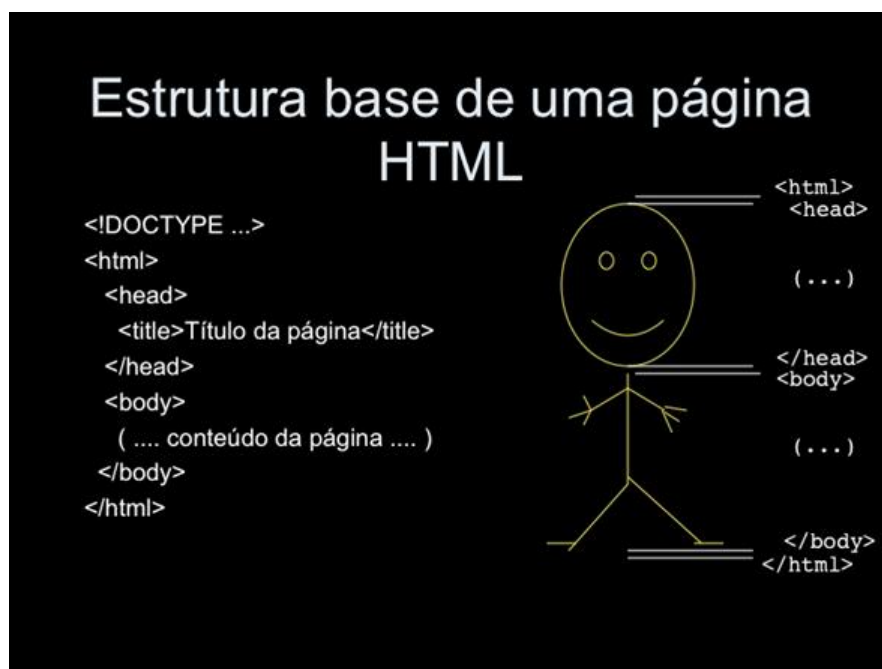


Figura 29: Estrutura de uma página HTML

#### 4.5.2 PHP

PHP é uma linguagem que possibilita a criação de páginas web dinâmicas, permitindo uma interação com o utilizador através de formulários, parâmetros da URL e links. A diferença do PHP para outras linguagens semelhantes a Javascript é que o código no caso do PHP é executado no servidor, enviando depois para o lado do cliente apenas HTML puro.

Desta forma é possível interagir com bases de dados e aplicações existentes no lado do servidor, tendo ainda a vantagem de não mostrar o código fonte para o lado do cliente, que é extremamente útil quando se lida com passwords ou outros dados confidenciais.

Uma diferença do PHP em relação a um script escrito em C ou Perl é que o código PHP fica inserido no próprio código HTML, enquanto que nos outros casos é necessário que o script gere todo o código HTML, ou então que o vá ler a um outro arquivo. [Lop16]

#### 4.5.3 JavaScript

JavaScript, também conhecida por JS, é uma linguagem de programação criada em 1995 por Brendan Eich enquanto trabalhava na Netscape Communications Corporation. Originalmente foi projetada para correr apenas no Netscape Navigator, o JavaScript tinha o objetivo de dar aos programadores formas de tornar certos processos de páginas web mais dinâmicos.

Após um ano a Microsoft tornou compatível o JavaScript no seu navegador, o que fez com que a linguagem se tornasse uma das tecnologias mais importantes e utilizadas na internet.

Embora a linguagem tenha esse nome, não se deve confundir o JavaScript com a linguagem Java.

O JavaScript ao invés de correr remotamente em servidores na internet, executa programas localmente - do lado do cliente. Ou seja, o JavaScript dá às páginas web a possibilidade de programação, transformação e processamento de dados enviados e recebidos, interagindo com a marcação e exibição de conteúdo da linguagem HTML e com a estilização desse conteúdo proporcionada pelo CSS nessas páginas.[Sil16]

#### 4.5.4 W3.CSS

O W3.CSS é um framework CSS recente e é desenvolvido e mantido pela W3Schools, além de ser totalmente gratuito, agiliza e facilita muito o desenvolvimento de páginas web elásticas, ou seja, adaptáveis a vários dispositivos.

De seguida uma lista com algumas das vantagens da utilização do W3.CSS:

- Mais simples e rápido que outros frameworks CSS.
- Fácil de usar e de aprender.
- Extremamente leve - usa apenas CSS padrão.
- Dispensa o uso de JavaScript ou jQuery.
- Foco em “Mobile First” - desenvolvido primeiro para celular e depois adaptado para o desktop.
- Design responsivo - feito para todos os dispositivos (telemóvel, desktop e tablet), mas especialmente para dispositivos móveis.
- Acelera aplicativos HTML móveis.
- Possui recursos diferenciados dos outros frameworks CSS.
- Desenvolvido com base em “Material Design” - padrão visual criado pelo Google com foco na simplicidade e na funcionalidade
- É gratuito e não precisa de licença para a sua utilização.[Fer16]

#### 4.5.5 Tecnologias Usadas

As tecnologias usadas para o desenvolvimento da aplicação web foram identificadas e descritas nas subsecções anteriores.

De um modo sucinto, a framework **W3.CSS** foi usada com o intuito de tornar o desenvolvimento da aplicação mais rápido permitindo não ter preocupações com as diferentes particularidades dos dispositivos pelos quais se pode aceder a uma página web e com as compatibilidades com os vários browsers.

De entre alguns templates disponíveis optou-se por escolher um dos mais simples para servir de base à aplicação. Esta opção deve-se ao objetivo de se ter uma plataforma intuitiva



para o utilizador, sendo que um template mais complexo podia dificultar a experiência dos utilizadores.

Na construção das várias páginas da aplicação no que se refere ao lado do cliente o **HTML** e o **JavaScript** foram os principais aliados, já do lado do servidor foi usado o **PHP**.

#### 4.5.6 Estrutura do Código/Pastas

A organização das pastas é simples, sendo que existem quatro pastas, uma com o já mencionado sparqlib, uma com os ficheiros HTML, uma com os ficheiros PHP e por fim uma com vários ficheiros como imagens, audios, videos e documentos de apoio à aplicação ou importados na mesma.

Nas pastas de HTML e PHP temos basicamente três pastas representativas das três classes da ontologia: "Pessoas", "Eventos" e "Documentos". Dentro dessas três pastas temos as operações de CRUD para cada uma delas, sendo que no caso das HTML apenas há os ficheiros adicionar, sendo as restantes operações de CRUD todas em PHP.

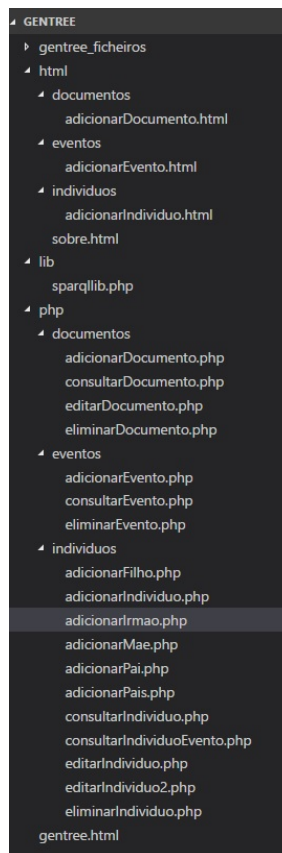


Figura 30: Estrutura das Pastas

#### 4.5.7 Template Usado - W3.CSS

O template usado foi escolhido de entre uma série de templates existentes na página do W3.CSS, o original pode ser consultado aqui [https://www.w3schools.com/w3css/tryw3css\\_templates\\_teal.htm](https://www.w3schools.com/w3css/tryw3css_templates_teal.htm).

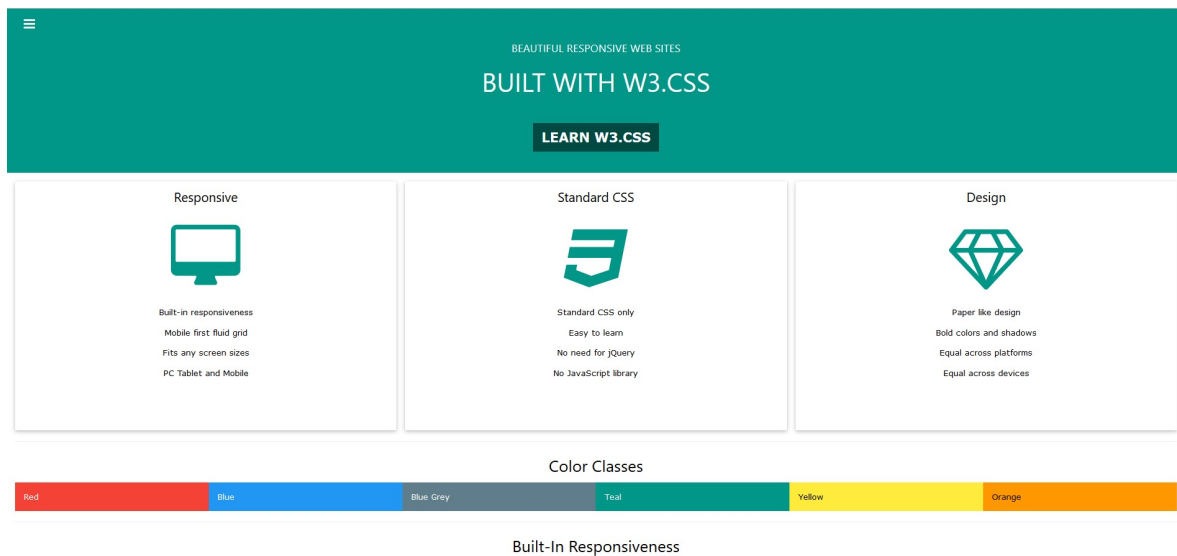


Figura 31: Template Usado

#### 4.5.8 Operações CRUD - SPARQL

Grande parte da aplicação web assenta nas operações de CRUD (acrónimo de Create, Read, Update e Delete), que traduzindo para português significa criar, consultar, atualizar e apagar, são estas as operações básicas de um sistema de base de dados.

Nas próximas páginas serão descritas as queries SPARQL usadas para realizar estas operações nas três principais classes da ontologia.

A estrutura do código que acompanha a query é simples, uma primeira parte que chama o sparqlib.php e define onde se deve conectar, a segunda parte é onde escrevemos a query, sendo que na terceira parte varia conforme a operação a ser realizada.

Exemplo da estrutura do código no caso de uma query de inserção:

```
error_reporting(E_ALL ^ E_NOTICE);
require_once("../lib/sparqlib.php");
$db = sparql_connect('http://localhost:7200/repositories/gentree_v4')
;
$sparql = " Aqui escreve-se a query";
header("application/sparql-update");
$service_url = 'http://localhost:7200/repositories/gentree_v4/
statements';
$curl = curl_init($service_url);
$curl_post_data = array('update' => $sparql);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, $curl_post_data);
curl_setopt($curl, CURLOPT_USERPWD, "admin:root");
curl_exec($curl);
curl_close($curl);
```

*Criar Pessoas*

A primeira query é a de adicionar uma nova pessoa no sistema, é um insert simples que pega nos dados colocados no formulário de adição de um novo individuo (será mostrado o formulário na secção "Aplicação - Exemplos") e os insere no grafo armazenado no GraphDB.

```

$sparql = "
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
INSERT DATA
{
    gentree:$query rdf:type gentree:Person.
    gentree:$query gentree:firstName '$pname' .
    gentree:$query gentree:secondName '$sname' .
    gentree:$query gentree:familyName '$fname' .
    gentree:$query gentree:lastName '$lname' .
    gentree:$query gentree:birthDate '$bdate' .
    gentree:$query gentree:sex '$gender' .
    gentree:$query gentree:address '$morada' .
    gentree:$query gentree:city '$cidade' .
    gentree:$query gentree:country '$pais' .
    gentree:$query gentree:educationDegree '$gacad' .
    gentree:$query gentree:email '$email' .
    gentree:$query gentree:fax '$fax' .
    gentree:$query gentree:height '$altura' .
    gentree:$query gentree:languagesSpoken '$linguas' .
    gentree:$query gentree:phone '$contacto' .
    gentree:$query gentree:physicalDescription '$descfisica' .
    gentree:$query gentree:placeOfEducation '$lform' .
    gentree:$query gentree:placeOfOccupation '$ltrab' .
    gentree:$query gentree:religion '$religiao' .
    gentree:$query gentree:weight '$peso' .
    gentree:$query gentree:occupation '$profissao' .
    gentree:$query gentree:interests '$interesse' .
    gentree:$query gentree:personalPhotoPath '$query' .
}";

```

*Criar Eventos*

No caso da query para adicionar eventos, ela varia conforme o evento que é pretendido inserir, no exemplo abaixo vemos a query para o caso do evento de casamento. O tipo de evento que se trata é incluído pela variável "event" na terceira linha do insert no exemplo abaixo.

```

$sparql = "
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
INSERT DATA
{
    gentree:$query gentree:eventHasPerson gentree:$person.
    gentree:$query gentree:eventHasPerson gentree:$person1.
    gentree:$query rdf:type gentree:$event.
    gentree:$query rdf:type gentree:Event.
    gentree:$query gentree:$date1 '$date'.
    gentree:$query gentree:$local1 '$local'.
    gentree:$query gentree:$description1 '$description'.
    gentree:$query gentree:$padrinho '$godfather'.
    gentree:$query gentree:$madrinha '$godmother'.
    gentree:$person rdf:type gentree:Person.
    gentree:$person1 rdf:type gentree:Person.
    gentree:$person gentree:isSpouseOf gentree:$person1.
}";

```

### *Criar Documentos*

Tal como no caso dos eventos, a query de inserir documentos também varia, menos é certo, mas mesmo assim as queries vão sendo alteradas ligeiramente conforme o documento a adicionar.

```
$sparql = "  
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>  
INSERT DATA  
  {  
    gentree:$query rdf:type gentree:Document.  
    gentree:$query rdf:type gentree:$doc.  
    gentree:$query gentree:docHasPerson gentree:$person.  
    gentree:$query gentree:photoPath '$newname' .  
  }";
```

*Listar Pessoas*

Após ser mostrado como se insere dados, será agora demonstrado como se pode consultar os dados da ontologia listando-os na aplicação web. Para isso foi usada a query "select" do SPARQL. Com a query seguinte obteve-se a lista de todas as pessoas do sistema ordenadas pelo id:

```
$sparql = "
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
SELECT ?person
  WHERE {
    ?person a gentree:Person.
  }
ORDER BY ?person";
```

Se o objetivo for além do id obter algumas informações associadas ao mesmo para ser mais fácil identificar cada instância, pode-se fazer uma tabela e colocar a informação pretendida seguindo um exemplo deste tipo:

```
$sparql = '
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
SELECT ?person ?firstname ?secondname ?surname ?lastname ?sex ?city ?
country
  WHERE {
    ?person a gentree:Person.
    ?person gentree:firstName ?firstname.
    OPTIONAL {?person gentree:secondName ?secondname.}
    OPTIONAL {?person gentree:familyName ?surname.}
    OPTIONAL {?person gentree:lastName ?lastname.}
    OPTIONAL {?person gentree:sex ?sex.}
    OPTIONAL {?person gentree:city ?city.}
    OPTIONAL {?person gentree:country ?country.}
  }
ORDER BY ?firstname';
```

### Listar Eventos

Na aplicação web existe um separador para cada tipo de evento, sendo o primeiro um separador geral com a lista de todos os eventos. Esse separador é alimentado pela query:

```
$sparql = '  
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>  
SELECT ?event ?firstname ?secondname ?surname ?lastname ?sex  
  WHERE {  
    ?person a gentree:Person.  
    ?event a gentree:Event.  
    ?event gentree:eventHasPerson ?person.  
    ?person gentree:firstName ?firstname.  
    OPTIONAL {?person gentree:secondName ?secondname.}  
    OPTIONAL {?person gentree:familyName ?surname.}  
    OPTIONAL {?person gentree:lastName ?lastname.}  
    OPTIONAL {?person gentree:sex ?sex.}  
  }  
ORDER BY ?event';
```



*Listar Documentos*

A estrutura da parte da consulta dos documentos é a mesma dos eventos, com separadores para cada tipo de documento, por isso mesmo, a query pouco se altera.

Há apenas a salientar que cada separador tem em tabelas separadas os documentos que estão associados às pessoas e os que estão associados a eventos, dando origem a duas queries por cada tipo de associação.

```
$sparql = '
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
SELECT ?audio ?path
  WHERE {
    ?person a gentree:Person.
    ?audio a gentree:Audio.
    ?audio gentree:docHasPerson ?person.
    ?audio gentree:audioPath ?path.
  }';
```

```
$sparql = '
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
SELECT ?audio ?path
  WHERE {
    ?event a gentree:Event.
    ?audio a gentree:Audio.
    ?audio gentree:docHasEvent ?event.
    ?audio gentree:audioPath ?path.
  }';
```

*Editar Pessoas*

A estrutura desta query consiste em pegar no id da instância que se pretende alterar, apaga-se a informação antiga e insere-se a nova nos campos respetivos daí ter também além do "Delete" e do "Insert" um "Where".

Desta forma não se perde o id original, nem a restante informação associada em termos de eventos, relações e documentos. Esta query é apenas para a parte dos dados pessoais.

```

$id = $_POST['ident'];
$sparql = "
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
DELETE
  {
    gentree:$id gentree:firstName '$firstname' .
    gentree:$id gentree:lastName '$lastname' .
    gentree:$id gentree:secondName '$secondname' .
    gentree:$id gentree:familyName '$surname' .
    gentree:$id gentree:city '$city' .
    gentree:$id gentree:country '$country' .
    gentree:$id gentree:birthDate '$birthdate' .
    gentree:$id gentree:address '$address' .
    gentree:$id gentree:email '$email' .
    gentree:$id gentree:phone '$phone' .
    gentree:$id gentree:fax '$fax' .
    gentree:$id gentree:educationDegree '$educationdegree' .
    gentree:$id gentree:placeOfEducation '$placeofeducation' .
    gentree:$id gentree:languagesSpoken '$languagesspoken' .
    gentree:$id gentree:occupation '$occupation' .
    gentree:$id gentree:placeOfOccupation '$placeofoccupation' .
    gentree:$id gentree:religion '$religion' .
    gentree:$id gentree:height '$height' .
    gentree:$id gentree:weight '$weight' .
    gentree:$id gentree:physicalDescription '$physicaldescription' .
    gentree:$id gentree:interests '$interests' .
  }
INSERT
  {
    gentree:$id gentree:firstName '$pname' .
    gentree:$id gentree:lastName '$lname' .
    gentree:$id gentree:secondName '$sname' .
    gentree:$id gentree:familyName '$fname' .
  }

```

```

gentree:$id gentree:address '$morada' .
gentree:$id gentree:city '$cidade' .
gentree:$id gentree:country '$pais' .
gentree:$id gentree:educationDegree '$gacad' .
gentree:$id gentree:email '$email' .
gentree:$id gentree:fax '$fax' .
gentree:$id gentree:height '$altura' .
gentree:$id gentree:languagesSpoken '$linguas' .
gentree:$id gentree:phone '$contacto' .
gentree:$id gentree:physicalDescription '$descfisica' .
gentree:$id gentree:placeOfEducation '$lform' .
gentree:$id gentree:placeOfOccupation '$ltrab' .
gentree:$id gentree:religion '$religiao' .
gentree:$id gentree:weight '$peso' .
gentree:$id gentree:occupation '$profissao' .
gentree:$id gentree:interests '$interesse' .
}
WHERE
{
OPTIONAL{gentree:$id gentree:firstName '$firstname'`}
OPTIONAL{gentree:$id gentree:lastName '$lastname'`}
OPTIONAL{gentree:$id gentree:secondName '$secondname'`}
OPTIONAL{gentree:$id gentree:familyName '$surname'`}
OPTIONAL{gentree:$id gentree:city '$city1'`}
OPTIONAL{gentree:$id gentree:country '$country1'`}
OPTIONAL{gentree:$id gentree:birthDate '$birthdate1'`}
OPTIONAL{gentree:$id gentree:address '$address1'`}
OPTIONAL{gentree:$id gentree:email '$email1'`}
OPTIONAL{gentree:$id gentree:phone '$phone1'`}
OPTIONAL{gentree:$id gentree:fax '$fax1'`}
OPTIONAL{gentree:$id gentree:educationDegree '$educationdegree1'`}
OPTIONAL{gentree:$id gentree:placeOfEducation '$placeofeducation1'`}
OPTIONAL{gentree:$id gentree:languagesSpoken '$languagesspoken1'`}
OPTIONAL{gentree:$id gentree:occupation '$occupation1'`}
OPTIONAL{gentree:$id gentree:placeOfOccupation '$placeofoccupation1'`}
OPTIONAL{gentree:$id gentree:religion '$religion1'`}
OPTIONAL{gentree:$id gentree:height '$height1'`}
OPTIONAL{gentree:$id gentree:weight '$weight1'`}

```

```
OPTIONAL{gentree:$id gentree:physicalDescription '
    $physicaldescription1'.}
OPTIONAL{gentree:$id gentree:interests '$interests1'.}
}";
```

Para além disso na edição de pessoas também é possível adicionar três relações de parentesco, pai, mãe e irmão.

No caso do pai e da mãe isso vai permitir ir buscar os parentescos já definidos com essas pessoas e inferir no indivíduo que está a ser editado todos esses parentescos. Exemplo: Ao adicionar um pai B à pessoa A se esse pai A também tiver um pai C, automaticamente vai ser inferido que a pessoa A tem como avô o pai C.

### Pai

```
$sparql = " PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree
#>
INSERT DATA {
    gentree:$id rdf:type gentree:Person.
    gentree:$id gentree:hasDad gentree:$dad.
}";
```

### Mãe

```
$sparql = " PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree
#>
INSERT DATA {
    gentree:$id rdf:type gentree:Person.
    gentree:$id gentree:hasMom gentree:$mom.
}";
```

Já no caso do irmão é preciso inserir desta forma porque não se consegue inferir essa relação automaticamente, tal como foi mencionado no capítulo da modelação e construção da ontologia.

Existem também inferências a serem feitas nesta operação de adição de irmãos, por exemplo ao inserir um irmão casado a uma pessoa é inferido que a esposa do irmão é cunhada dessa pessoa.

```
$sparql = " PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree
#>
INSERT DATA {
    gentree:$id rdf:type gentree:Person.
    gentree:$id gentree:isSiblingOf gentree:$brother.
}";
```

### *Editar Eventos e Documentos*

Nestes dois casos não foi implementada qualquer forma de editar o já existente no sistema.

### *Eliminar Pessoas*

Sabendo o id a eliminar, a query tem de apagar tudo o que está associado ao mesmo, há no entanto no sistema uma nuance no que toca a regras para ser possível correr esta query. Caso haja eventos ou documentos associadas tentar eliminar uma pessoa isso não é permitido.

```
$id = $_POST['person'];  
$sparql = "  
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>  
DELETE WHERE  
  {  
    gentree:$id ?p ?o.  
  }";
```

### *Eliminar Eventos*

No caso de eliminar eventos o processo é o mesmo excepto a parte das condicionantes, basicamente utiliza-se o id do evento seleccionado para fazer delete a toda a informação associada.

```
$id = $_POST['event'];  
$sparql = "  
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>  
DELETE WHERE  
  {  
    gentree:$id ?p ?o.  
  }";
```

### Eliminar Documentos

Mesma query que no caso anterior, neste caso há uma diferença no que toca a eliminar os documentos, já que a query apenas apaga a informação passada para a ontologia e não o documento em si.

```
$id = $_POST['doc'];
$sparql = "
PREFIX gentree: <http://www.di.uminho.pt/mei2017/gentree#>
DELETE WHERE
{
    gentree:$id ?p ?o.
}";
```

Devido a esse facto é preciso apagar os documentos do local onde estão guardados de outra forma, para este procedimento recorreu-se a código PHP. A seguir o exemplo de como proceder para eliminar um ficheiro áudio.

```
$dir = '../..../gentree_ficheiros/Audio';
$dirHandle = opendir($dir);
while ($file = readdir($dirHandle))
{
    if(utf8_encode($file)==$path)
    {
        unlink($dir."/".$file);
    }
}
closedir($dirHandle);
```



---

## APLICAÇÃO - EXEMPLOS

---

Este capítulo será dedicado à visualização de imagens da aplicação usando exemplos de teste de modo a tornar o funcionamento deste sistema mais perceptivo.

### 5.1 INTERFACE PRINCIPAL

A página principal da aplicação alberga praticamente atalhos para todas as funcionalidades da mesma. Dividido também em três grandes grupos tal como foi feito nas outras partes desta dissertação.

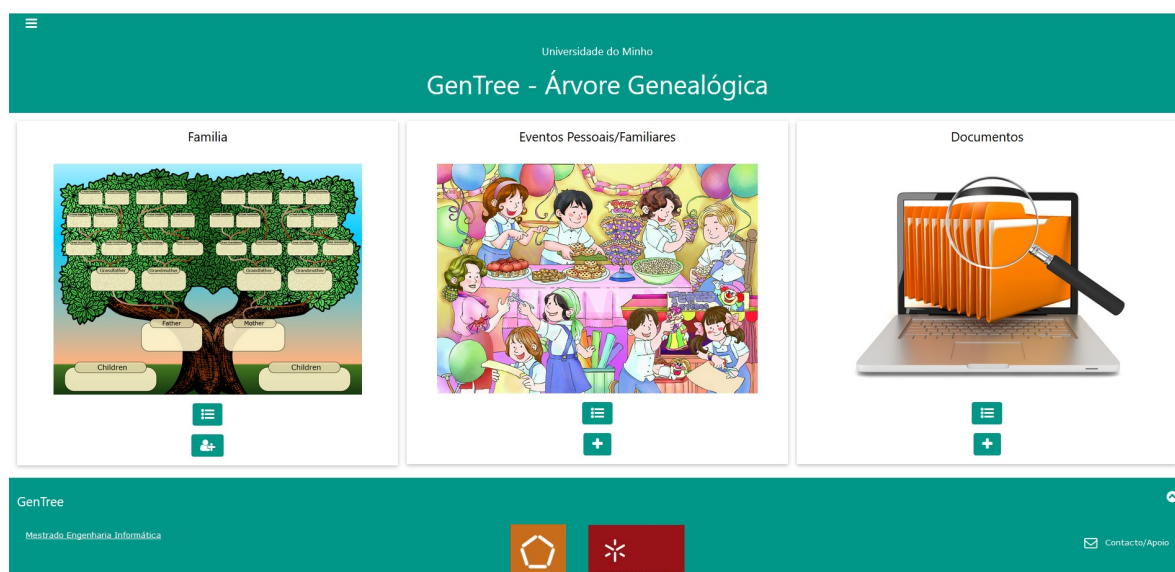


Figura 32: Página Principal

Sob cada imagem associada a um desses grupos encontram-se atalhos para adicionar e listar as instâncias. Outro caminho é o botão no canto superior esquerdo que permite também aceder a atalhos semelhantes aos referidos anteriormente. No fundo da página

existe algumas referências que contextualizam o âmbito da realização deste projeto.

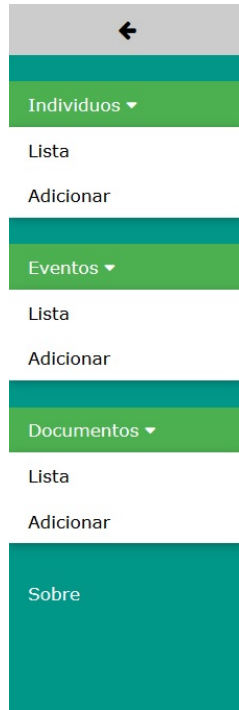


Figura 33: Menu de Atalhos

## 5.2 FUNCIONALIDADES RELATIVAS A PESSOAS

### 5.2.1 Lista de Pessoas

Nesta página temos a tabela com todas as pessoas que inserimos da nossa família, a página possui uma caixa de selecção logo após o cabeçalho que permite escolher uma pessoa em particular e depois disso optar por uma das três opções que vemos representadas pelos botões abaixo dessa caixa de selecção.

- Ver indivíduo - Encaminha para uma página onde se vê os detalhes da pessoa seleccionada;
- Ver eventos - Encaminha para uma página onde se pode visualizar os eventos associados à pessoa seleccionada;
- Eliminar indivíduo - Apaga a pessoa seleccionada do sistema (caso cumpra as regras referidas anteriormente).

The screenshot shows the 'GenTree - Árvore Genealógica' interface. At the top, there is a header with the university name and a search bar. Below the header, there is a dropdown menu labeled 'Escolha uma Pessoa'. Underneath, there are three buttons: 'Ver Indivíduo', 'Ver Eventos', and 'Eliminar Indivíduo'. The main part of the interface is a table listing family members with columns for ID, First Name, Second Name, Surnames, Last Name, and Gender.

ID Indivíduo	Primeiro Nome	Segundo Nome	Apelidos	Ultimo Nome	Género
Adriana_Sofia_Gomes_Ribeiro_Moreira_Santos	Adriana	Sofia	Gomes Ribeiro Moreira	Santos	F
Adriano_Teixeira	Adriano			Teixeira	M
Alberto_Ribeiro	Alberto			Ribeiro	M
Alcides_Moreira	Alcides				M
Alexandra_Ribeiro_Araújo	Alexandra		Ribeiro	Araújo	F
André_Silva_Moreira	André		Silva	Moreira	M
André_Ribeiro_Araújo	André		Ribeiro	Araújo	M
António_Luis_Moreira_Santos	António	Luis	Moreira	Santos	M
Arminda_Moreira	Arminda			Moreira	F
Carmo_Ribeiro	Carmo			Ribeiro	F
Célia_Ribeiro	Célia			Ribeiro	F

Figura 34: Listagem de Pessoas da Família

### 5.2.2 Detalhes da Pessoa

Seleccionando uma pessoa e abrindo a sua página, tem-se acesso à informação pessoal da mesma e às relações de parentesco agrupadas em vários separadores.



Figura 35: Dados da Pessoa

Nesta imagem vê-se o separador correspondente aos pais da pessoa seleccionada.



Figura 36: Pais da Pessoa

### 5.2.3 Adicionar

O formulário para inserir novas pessoas no sistema divide-se em 2 partes, uma parte com os dados típicos de muitos formulários sobre os dados pessoais e uma segunda parte para se inserir uma foto de perfil.

No caso da foto importada, irá ser associada ao perfil da pessoa criada e guardada numa pasta local, caso não seja inserida nenhuma foto de perfil no seu lugar quando for aberto o perfil da pessoa haverá uma imagem prédefinida do sistema.

A imagem tem um zoom considerável para se ver o formulário completo.

Universidade do Minho  
GenTree - Árvore Genealógica

Adicionar Indivíduo

Dados Pessoais		Dados Acadêmicos	
Primeiro Nome		Fax	
Segundo Nome		Grau Académico	
Outros Apelidos		Local de Formação	
Último Nome		Profissão	
Data de Nascimento		Local de Trabalho	
Dia [1] Mês [1] Ano [1]		Línguas Faladas	
Sexo	<input checked="" type="radio"/> Masculino <input type="radio"/> Feminino	Religião	
Marital		Altura	
Cidade		Peso	
País		Descrição Física	
Etnia		Interesses	
Telefone/Fax			

Foto (Máx: 1 MB)  
Explorar Nenhum ficheiro selecionado.

GenTree

Figura 37: Inserir Pessoa

### 5.2.4 Editar

No fundo da página da informação da pessoa há um botão (o mais escuro dos dois) que quando é carregado desbloqueia a edição dos dados, mas isso apenas acontece nos campos que foram definidos como editáveis, a data de nascimento por exemplo não pode ser alterada.

Ao premir o botão de editar além dos campos passarem a ser editáveis também o botão de guardar passa a estar ativo para no final de se realizarem as alterações pretendidas ser possível gravar as mesmas.



Figura 38: Editar Pessoa

Nesta página entra também um aspeto crucial de todo o sistema que é o de adicionar as três relações de parentesco possíveis, a partir das quais serão inferidas as restantes.

- Pai;
- Mãe;
- Irmão.

O funcionamento da edição é igual ao dos dados pessoais, é necessário ativar os campos clicando no botão de edição que se encontra no fundo da página e depois seleciona-se uma das relações para se associar uma pessoa, assim que seja feita a escolha da pessoa grava-se essa associação.

Para repetir o procedimento para outras relações é necessário voltar a abrir o perfil da pessoa.

⚠  
Não adicione relações que já estejam definidas!

Adicionar Irmão

**Irmão**  
Escolha uma Pessoa

📄

Adicionar Pai

**Pai**  
Escolha uma Pessoa

📄

Adicionar Mãe

**Mãe**  
Escolha uma Pessoa

📄

✍️

Figura 39: Adicionar Relações de Parentesco

Adicionar estes relacionamentos correctamente é de extrema importância, já que sem eles não é possível serem feitas as inferências das restantes relações familiares. Conforme se vai construindo a árvore, adicionando pessoas e respetivas relações de pai, mãe e irmãos, o número de relações de parentesco assumidas automaticamente vai aumentando significativamente.

### 5.2.5 Eliminar

Como já foi referido, para se eliminar uma pessoa ela não pode ter eventos nem documentos associados, neste caso a pessoa que se tentou eliminar não cumpria essas regras logo é apresentada a mensagem que se vê na imagem.



Figura 40: Mensagem associada à impossibilidade de se eliminar uma pessoa



## 5.3 FUNCIONALIDADES RELATIVAS A EVENTOS

### 5.3.1 Lista de Eventos

Na lista de eventos vemos uma tabela com todos os eventos presentes no sistema. A primeira coluna é o ID do evento, essa informação é útil no caso de se usar a caixa de selecção saber a que evento cada ID pertence.



Universidade do Minho

## GenTree - Árvore Genealógica

Todos Adopção Batizado Nascimento Aniversário Comunhão Casamento Divórcio Emigração Noivado Serviço Militar Naturalização Reforma Óbito

Escolha um Evento

Eliminar Evento

ID	Primeiro Nome	Segundo Nome	Apelidos	Último Nome	Género
birth_0001	Frederico		Moreira	Ribeiro	M
birth_0002	Ricardo		Moreira	Ribeiro	M
birth_0003	José	Alberto	Gomes	Ribeiro	M
birth_0004	Maria	Isabel	Silva	Moreira	F
birth_0005	Natália	Maria	Silva	Moreira	F
birth_0006	Tânia			Silva	F
birth_0007	Joaquim	Alcides	Silva	Moreira	M
birth_0008	Luis	Miguel	Gomes Ribeiro Moreira	Santos	M
birth_0009	Adriana	Sofia	Gomes Ribeiro Moreira	Santos	F
birthday_0001	Frederico		Moreira	Ribeiro	M

Figura 41: Lista de Eventos

Esta página contém mais separadores, cada um deles refere-se a um tipo de evento diferente, na imagem a seguir vê-se o separador das emigrações.



Universidade do Minho

## GenTree - Árvore Genealógica

Todos Adopção Batizado Nascimento Aniversário Comunhão Casamento Divórcio Emigração Noivado Serviço Militar Naturalização Reforma Óbito

Primeiro Nome	Segundo Nome	Apelidos	Último Nome	Género	País	Data
Maria	Isabel	Silva	Moreira	F	Alemanha	1976
Joaquim		Silva	Moreira	M	Alemanha	1976
Joaquim		Silva	Moreira	M	Alemanha	1976
Maria	Conceição	Silva	Araújo	F	Alemanha	1976
Natália	Maria	Silva	Moreira	F	Alemanha	1976
José			Ribeiro	M	Alemanha	
Laurinda		Gomes	Carvalho	F	Alemanha	
Maria	Conceição	Gomes	Carvalho	F	Alemanha	
Miquelina			Araújo	F	Brasil	

GenTree

Figura 42: Lista de Emigrações

### 5.3.2 Adicionar

No processo de adição de eventos temos de associar o evento a uma pessoa, há eventos que podem necessitar de duas pessoas como é o caso do casamento, quando isso sucede fica desbloqueada a caixa de selecção para a segunda pessoa.

Os campos possíveis de serem preenchidos também ficam bloqueados ou desbloqueados conforme o evento que se pretende criar.

Universidade do Minho

## GenTree - Árvore Genealógica

Adicionar Evento

Evento: Escolha uma Pessoa Pessoa 1: Escolha uma Pessoa Pessoa 2: Escolha uma Pessoa

Data do Evento  
 Dia 1 Mês 1 Ano 2017

Local do Evento

Descrição do Evento

Número de Pessoas Presentes

Padrinho

Madrinha

Duração

País

GenTree

Mestrado, Engenharia Informática

UNIVERSIDADE DO MINHO

Figura 43: Adicionar Eventos

### 5.3.3 Eliminar

Para eliminar um evento basta usar a caixa de selecção no separador com todos os eventos para seleccionar o evento pretendo e carregar no botão que está um pouco abaixo.

Ao eliminar um evento faz ligação a uma página que nos indica os dados do evento que acabou de ser apagado. Há linhas não estarão preenchidas visto os eventos não terem sempre as mesmas informações associadas.



Figura 44: Eliminar Evento

## 5.4 FUNCIONALIDADES RELATIVAS A DOCUMENTOS

Os ficheiros importados no sistema irão para pastas locais do computador onde está a correr o servidor disponibilizado pelo XAMPP, ainda foi estudado passar todo o sistema para a cloud e com esse intuito foi testada a utilização de uma plataforma do género do Dropbox para armazenar os ficheiros mas essa ideia acabou por não seguir em frente.

Foram criadas várias pastas de modo a que cada tipo de documento tivesse a sua própria pasta, as operações de adicionar e eliminar que serão descritas nos pontos seguintes agem também sobre estas pastas além das informações passadas ao GraphDB.

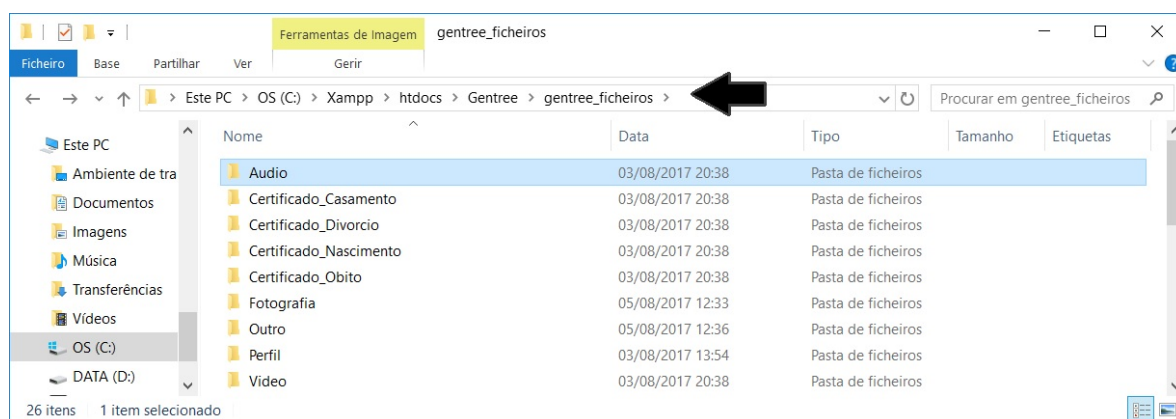


Figura 45: Pastas para os Documentos

### 5.4.1 Lista de Documentos

Na lista de documento não vemos uma tabela com todos os documentos tal como aconteceu nas pessoas e nos eventos, apenas existe a caixa de selecção e o botão para eliminar o documento seleccionado. Aqui optou-se por ter apenas nos separadores a lista dos documentos correspondentes a cada um tipo de documento.

Ao seleccionar o documento, o mesmo será aberto noutra separador do browser, em certo tipo de ficheiros aparecerá em vez disso a janela para decidir se abre ou se transfere o documento para o seu computador.



Figura 46: Lista Documentos

Na imagem seguinte vê-se o caso do separador das fotografias, em todos os separadores existe uma tabela para os documentos associados às pessoas e uma tabela para os documentos associados aos eventos.



Figura 47: Lista de Fotografias

### 5.4.2 Adicionar

O formulário para adicionar documentos também tem algumas nuances conforme as opções que vai escolhendo, por exemplo se escolher associar o documento a uma pessoa, a caixa de selecção com os eventos permanecerá bloqueada ao contrário da caixa relativa às pessoas que ficará desbloqueada.

Os ficheiros importados irão para pastas locais do computador onde está a correr o servidor disponibilizado pelo XAMPP.

The screenshot shows the 'Adicionar Documento' (Add Document) form in the GenTree application. The header is green with the text 'Universidade do Minho' and 'GenTree - Árvore Genealógica'. The form is titled 'Adicionar Documento' and contains several sections:

- Associar Documento a:** A dropdown menu with the text 'Escolha uma Opção'.
- Evento:** A text input field.
- Pessoa:** A text input field.
- Tipo de documento:** A dropdown menu with the text 'Escolha o Tipo de Documento'.
- Ficheiro (Máx: 20 MB):** A file upload section with a button labeled 'Explorar...' and the text 'Nenhum ficheiro selecionado.' Below this is a green button with a document icon.

Figura 48: Adicionar Documento

### 5.4.3 Eliminar

No que toca a eliminar, tal como foi referido anteriormente basta seleccionar o documento que se pretende apagar na caixa de selecção existente no separador "Todos" e carregar no botão de eliminar. No caso da imagem percebe-se que o documento foi apagado com sucesso.



Figura 49: Eliminar Documento

## 5.5 SOBRE

Página relativa às informações e créditos da plataforma.



Figura 50: Sobre



---

## CONCLUSÃO

---

### 6.1 CONCLUSÕES

No final desta tese de mestrado o pretendido era obter-se uma aplicação web de genealogia suportada por uma ontologia em que fosse possível fazer as várias operações de CRUD necessárias para se criar e manter uma árvore geneológica.

Inicialmente houve um estudo do que já existe e de como se deveria proceder para o desenvolvimento deste sistema.

A primeira parte foi desenvolver a ontologia, fase essa que se revelou bastante complicada, sendo que o desenvolvimento resultou em avanços e recuos enquanto se trabalhava no papel, na altura em que se passou para o Protégé e se começou a testar o que foi pensado no papel ficou mais claro o que seria necessário para se obter o resultado pretendido.

Houve contudo o problema da inferência dos irmãos que fez perder bastante tempo sem nenhuma solução alcançada provocando até algum desânimo visto ser um problema numa das principais relações de parentesco.

Na segunda fase foi necessário entender e configurar o GraphDB para dar respostas às necessidades da ontologia e da aplicação web que era pretendido criar, neste ponto houve alguma troca de email's com o apoio técnico do GraphDB de modo a obter alguma ajuda, obtendo sempre um feedback positivo da parte deles.

Com a parte da ontologia fechada, foi preciso arrancar com a aplicação web, o desafio nesta fase foi criar as operações de CRUD e fazer com que a ligação com o GraphDB fosse possível, até encontrar o sparqlib tinha-se um problema em mãos, felizmente o professor José Carlos Ramalho tinha conhecimento de um projeto onde essa interação tinha sido

usada e forneceu-me acesso ao mesmo.

A partir daqui a criação das queries necessárias com maior ou menor dificuldade acabou por ser realizada.

Houve ainda uma fase de testes para verificar se as operações estavam a ser realizadas como pretendido. Paralelamente a todas estas fases foi sendo escrita esta dissertação.

No geral os objetivos foram cumpridos, não da forma ideal já que surgiram alguns obstáculos, mas de maneira a que neste momento exista uma base a nível da ontologia e da aplicação web que dá resposta ao que inicialmente foi delineado como objetivo.

Posto isto, há alguns aspetos que podem ser melhorados, alguns deles irão ser referido na secção seguinte.

## 6.2 TRABALHO FUTURO

Ao fim de vários meses a trabalhar neste projeto é claro para mim que há ainda desenvolvimentos que podem dar mais valor ao sistema e ajudar a um dia tornar este sistema acessível a ser utilizado por qualquer pessoa que assim o pretenda.

De seguida alguns pontos que desenvolvidos seriam de grande mais valia:

- Criar login para possibilitar a utilização de vários utilizadores num sistema centralizado. Esta parte passaria por atribuir a cada ontologia um ID e posteriormente associar a cada uma das contas criadas. O GraphDB possibilita esta multiplicação de grafos;
- Integrar na aplicação web a questão da constituição de "agregados familiares" que se encontra já pronta na parte da ontologia;
- Desenvolver um método alternativo para inferir a relação de "irmãos" compatível com os motores de inferência, já que neste momento isso está a ser feito por inserção manual do utilizador o que não é o ideal;
- Visualização gráfica, tipo árvore, para melhor percepção dos parentescos entres os elementos da família.

Podem haver outras situações a serem trabalhadas, sendo que, de momento estas parecem ser as mais adequadas a serem desenvolvidas com base no trabalho já realizado.

---

## BIBLIOGRAFIA

---

- [AB03] Mauricio B. Almeida and Marcello P. Bax. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. <http://revista.ibict.br/ciinf/article/viewFile/984/1023/1082>, 2003.
- [Anc16] Ancestry. Página do ancestry. <http://trees.ancestry.com/>, 2016.
- [Con16] Conceito.de. Genealogia. <http://conceito.de/>, 2016.
- [Cre16] Crestleaf. Página do crestleaf. <http://crestleaf.com/>, 2016.
- [Ech16] Family Echo. Página do family echo. <http://www.familyecho.com/>, 2016.
- [Fam16a] FamilySearch. Página do familysearch. <https://familysearch.org/>, 2016.
- [Fam16b] FamilyTreeNow. Página do familytreenow. <http://www.familytreenow.com/>, 2016.
- [Fer16] Angelo Luis Ferreira. W3.css o framework que você precisa conhecer! <http://visualdicas.blogspot.pt/2016/08/w3css-o-framework-que-voce-precisa.html>, 2016.
- [GED96] GEDCOM. Modelo de dados gedcom. <http://homepages.rootsweb.ancestry.com/pmc-bride/gedcom/55gctoc.htm>, 1996.
- [Gen16a] Geneanet. Página do geneanet. <http://pt.geneanet.org/>, 2016.
- [Gen16b] Geni. Página do geni. <https://www.geni.com/family-tree/start>, 2016.
- [HTM17] HTML. World wide web consortium - html. <https://www.w3.org/html/>, 2017.
- [IB14] Seiji Isotani and Ig Ibert Bittencourt. Engenharia de ontologias. <http://ceweb.br/livros/dados-abertos-conectados/capitulo-4/>, 2014.
- [Inf16] Infoescola. Árvore geneológica. <http://www.infoescola.com/sociedade/arvore-genealogica/>, 2016.
- [Lop16] S. Lopes. O que é php. <https://www.oficinadanet.com.br/artigo/659/o-que-e-php>, 2016.

- [MAo7] Edison Andrade Martins Morais and Ana Paula L. Ambrósio. Ontologias: conceitos, usos, tipos, metodologias, ferramentas e linguagens. [http://www.portal.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF\\_001-07.pdf](http://www.portal.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_001-07.pdf), 2007.
- [Mac11] José Alexandre Macedo. Graph database: Soluções na literatura e implementações disponíveis. <http://jamacedo.com/2011/07/graph-database/>, 2011.
- [mp16] Find my past. <https://www.findmypast.com/family-tree>, 2016.
- [MyH16] MyHeritage. Página do myheritage. <https://www.myheritage.com.pt/>, 2016.
- [Ont16] Ontotext. Graphdb. <http://graphdb.ontotext.com/>, 2016.
- [OWL13] OWL. World wide web consortium - owl. <https://www.w3.org/2001/sw/wiki/OWL>, 2013.
- [RDF14] RDF. World wide web consortium - rdf, 2014.
- [Roc15a] Manuel Rocha. Vamos ligar e partilhar? – introdução ao rdf. <https://pplware.sapo.pt/internet/ligar-partilhar-introducao-ao-rdf/>, 2015.
- [Roc15b] Manuel Rocha. Vamos ligar e partilhar? – introdução ao sparql. <https://pplware.sapo.pt/internet/ligar-partilhar-introducao-ao-sparql/>, 2015.
- [Roc16a] Manuel Rocha. Vamos ligar e partilhar? – introdução às ontologias. <https://pplware.sapo.pt/internet/ligar-partilhar-introducao-as-ontologias>, 2016.
- [Roc16b] Manuel Rocha. Vamos ligar e partilhar? – visão geral sobre owl. <https://pplware.sapo.pt/internet/vamos-ligar-e-partilhar-visao-geral-sobre-owl/>, 2016.
- [Rod16] Joel Rodrigues. Html básico - códigos html. <http://www.devmedia.com.br/html-basico-codigos-html/16596>, 2016.
- [RTSGo8] Sandro Rautenberg, José L. Todesco, Andrea V. Steil, and Fernando A. O. Gauthier. Uma metodologia para o desenvolvimento de ontologias. <http://revistas.unicentro.br/index.php/RECEN/article/viewFile/711/884>, 2008.
- [Sil16] Giancarlo Silva. O que é e como funciona a linguagem javascript? <https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>, 2016.

- [SPA<sub>13</sub>] SPARQL. World wide web consortium - sparql.  
[https://www.w3.org/2009/sparql/wiki/Main\\_Page](https://www.w3.org/2009/sparql/wiki/Main_Page), 2013.