

Escola de Engenharia
Universidade do Minho

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

PROCESSAMENTO E REPRESENTAÇÃO DE CONHECIMENTO

CryptoNav

Octávio Maia A71369

17 de Junho de 2018

Conteúdo

1	Introdução	3
2	Especificação	4
2.1	Classes	4
2.2	Object properties	6
2.3	Data properties	7
3	Povoamento	8
4	Demonstração	9
4.1	Representação em grafos	9
4.2	Queries SPARQL	13
5	Conclusão	15

Lista de Figuras

1	Classes presentes na ontologia.	5
2	Anotações da classe <i>DistributionScheme</i>	5
3	Anotações da classe <i>ICO</i>	5
4	Anotações da classe <i>POS</i>	5
5	Object properties presentes na Ontologia.	6
6	Data properties presentes na Ontologia.	7
7	Diversos indivíduos presentes na nossa Ontologia.	8
8	Grafo representativo de todas as <i>Cryptocurrency</i>	9
9	Grafo representativo de todas as <i>Cryptocurrency</i> baseadas em POS.	9
10	Grafo representativo de todas as <i>Cryptocurrency</i> baseadas em POW.	10
11	Grafo representativo de todos os algoritmos e as <i>Cryptocurrency</i> que os utilizam. . . .	10
12	Grafo representativo da distribuição de todas as <i>Cryptocurrency</i>	11
13	Grafo representativo de toda a informação presente numa <i>Cryptocurrency</i> . (neste exemplo foi utilizado a moeda <i>XRP</i> como exemplo, visto ser <i>POS</i> o que leva a ter menos informação para representar.)	12

1 Introdução

CryptoNav surge no âmbito do perfil de Processamento e Representação de Conhecimento.

O objetivo principal deste projeto consiste no desenvolvimento de uma ontologia para um domínio escolhido pelos alunos.

Também é esperado o desenvolvimento de queries em *SPARQL*¹ de modo a exemplificar a informação que pode ser extraída da ontologia, bem como a criação de um website para a navegação na mesma.

Para este projeto, foi escolhido como dataset as *Cryptocurrencies*² e a tecnologia por trás da mesma, a *Blockchain*³.

Assim sendo, o nome da ontologia criada é *OACC - Ontology of a Cryptocurrency*.

¹<https://www.w3.org/TR/rdf-sparql-query/>

²<https://en.wikipedia.org/wiki/Cryptocurrency>

³<https://en.wikipedia.org/wiki/Blockchain>

2 Especificação

2.1 Classes

De modo a proceder ao desenvolvimento da ontologia, foi necessário proceder à especificação da mesma.

Começou-se por definir as classes necessárias para a nossa ontologia:

- **Cryptocurrency** - Uma instância de *Cryptocurrency*.
- **Creator** - Criador de uma certa *Cryptocurrency*.
- **DistributionScheme** - Esquema de distribuição das moedas geradas (com 4 subclasses).
 - **Fork** - Um *fork* ocorre quando a *blockchain* se divide em dois, através da criação de uma nova moeda, baseada na original. Quando isto ocorre, passa a haver replicação das moedas existentes na versão "antiga" da *blockchain* para a nova versão, num rácio definido pelo criador (normalmente este rácio é de 1:1).
 - **ICO** - *Initial coin offering* é um método recente de novos projetos de *Cryptocurrency* ganharem algum capital para investirem em certos features, como a listagem sem *exchanges*, etc. Para este fim, fazem uma venda de X moedas, tendo cada uma um preço fixo (normalmente este preço é expresso em *Bitcoin* ou *Ether*).
 - **Premine** - A pré-mineração de uma certa *Cryptocurrency* é baseado na mineração ou alocação prévia da moeda por alguns membros da equipa desenvolvedora, antes do lançamento ao público.
 - **None** - Ocorre quando não existe *ICO/Premine/Fork*, ou seja, um lançamento justo da moeda para todos os participantes.
- **HashingAlgorithm** - Algoritmo de uma dada *Cryptocurrency*. Este algoritmo é utilizado para a mineração de novas moedas na rede, apenas se aplica a moedas baseadas no modelo *Proof-of-work*. Exemplos de algoritmos são *SHA-256*, *Scrypt*, *Lyra2rev*, *Ethash*, *Equihash*, etc.
- **ProtectionScheme** - Esquema de proteção e validação da rede.
 - **POS** - *Proof-of-stake* é baseado na validação, através da escolha aleatória (mas influenciável) de quem irá gerar o novo bloco na rede. Em moedas *POS* as mesmas não são minadas, mas sim cunhadas. Este tipo de validação requer menos poder computacional que *POW*, mas é influenciável no facto de quem possuir mais moedas, tem mais probabilidade de ser escolhido para cunhar a próxima.
 - **POW** - *Proof-of-work* é baseado na validação, através de um algoritmo. Neste esquema, os *miners* alocam poder computacional para resolver os algoritmos, sendo recompensados pela validação da rede e resolução do algoritmo via um bloco, sendo que este contém moedas.

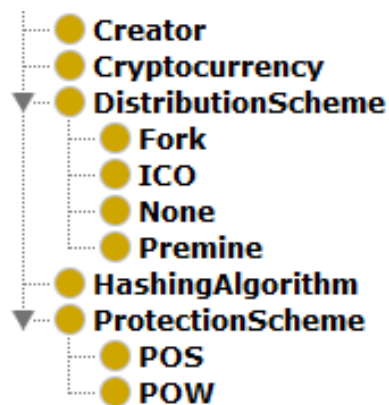


Figura 1: Classes presentes na ontologia.

Annotations: DistributionScheme

Annotations +

desc:about [language: en]
How the initial distribution of coins should be conducted between miners and non-miners.

Figura 2: Anotações da classe *DistributionScheme*.

Annotations: ICO

Annotations +

desc:about [language: en]
An ICO (Initial Coin Offering) is used by startups to bypass the rigorous and regulated capital-raising process required by venture capitalists or banks. The main purpose is to get sufficient hard-currency in exchange for an amount of Cryptocurrency.

desc:title [language: en]
Initial Coin Offering

Figura 3: Anotações da classe *ICO*.

Annotations: POS

Annotations +

desc:about [language: en]
A method of securing a cryptocurrency peer-to-peer network through requesting users to show ownership of a certain amount of currency.

desc:title [language: en]
Proof of Stake

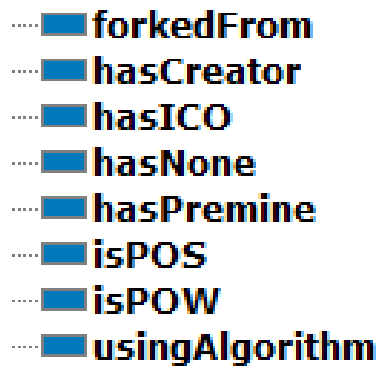
Figura 4: Anotações da classe *POS*.

2.2 Object properties

A próxima fase foi a declaração das *Object properties*.

De modo a simplificar a ontologia apenas foram declaradas as seguintes propriedades:

- **forkedFrom** - Indica se uma *Cryptocurrency* é um *fork* de outra.
- **hasCreator** - Indica o criador de uma *Cryptocurrency*.
- **hasICO** - Indica se uma dada *Cryptocurrency* teve um *ICO*.
- **hasNone** - Indica se uma dada *Cryptocurrency* não teve *ICO* nem *pre-mine*.
- **hasPremine** - Indica se uma dada *Cryptocurrency* teve *pre-mine*.
- **isPOS** - Indica se uma dada *Cryptocurrency* utiliza *Proof-of-stake*.
- **isPOW** - Indica se uma dada *Cryptocurrency* utiliza *Proof-of-work*.
- **usingAlgorithm** - Indica o algoritmo que uma *Cryptocurrency* utiliza.



..... **forkedFrom**
..... **hasCreator**
..... **hasICO**
..... **hasNone**
..... **hasPremine**
..... **isPOS**
..... **isPOW**
..... **usingAlgorithm**

Figura 5: Object properties presentes na Ontologia.

2.3 Data properties

Após a declaração dos *Object properties* apenas nos resta a declaração das *Data Properties*.

De modo a representar cada *Cryptocurrency* de maneira fiel foram declaradas as seguintes propriedades:

- **about** - Breve descrição de uma *Cryptocurrency*.
- **blockreward** - Recompensa que cada bloco contém.
- **blocktime** - Tempo em que novos blocos são colocados na rede.
- **circulatingsupply** - Quantidade de moedas de uma dada *Cryptocurrency* em circulação.
- **founded** - Data de criação da *Cryptocurrency*.
- **icoamount** - Quantidade de moedas vendidas em *ICO*.
- **maxsupply** - Quantidade máxima que irá existir de uma dada *Cryptocurrency*.
- **name** - Nome da uma *Cryptocurrency*.
- **networkdif** - Dificuldade da rede de uma *Cryptocurrency*.
- **networkhashrate** - Hashrate de uma dada *Cryptocurrency*.
- **premineamount** - Quantidade de moedas geradas em *pre-mine*.
- **price** - Preço de uma dada *Cryptocurrency*.
- **projectwhitepaper** - Link para o whitepaper de uma *Cryptocurrency*.
- **symbol** - Logótipo de uma *Cryptocurrency*.
- **tag** - Identificador de uma *Cryptocurrency*.
- **website** - Website de uma *Cryptocurrency*.

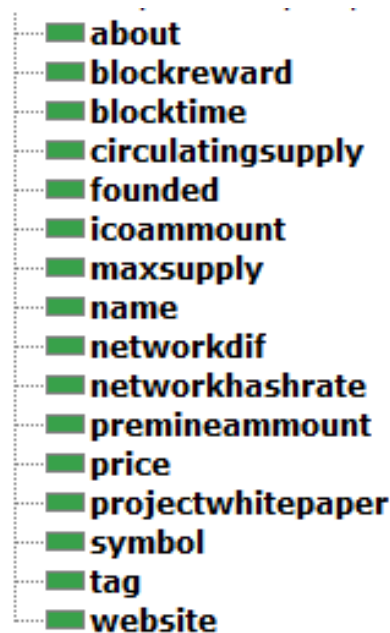


Figura 6: Data properties presentes na Ontologia.

3 Povoamento

Após feita o levantamento de requisitos e a especificação da ontologia, apenas falta o preenchimento do dataset.

Este foi preenchido com 21 *Cryptocurrencies*, sendo que cada uma delas tem o seu próprio *Creator* e *Algorithm*.

Alguns exemplos de *Cryptocurrency* inseridas:

- Bitcoin
- Bitcoin Cash
- Ethereum
- Decred
- Loki
- Monero
- etc.

Bem como algoritmos:

- SHA-256
- Ethash
- Equihash
- X11
- X16R
- Cryptonight
- etc.

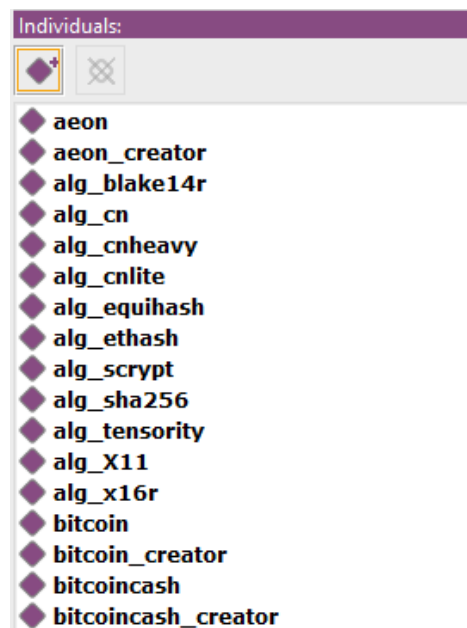


Figura 7: Diversos indivíduos presentes na nossa Ontologia.

4 Demonstração

De modo a demonstrar a informação recolhida foi criado um website utilizando *NodeJS*, *PUG* e outras ferramentas exploradas ao longo desta unidade curricular.

De modo a permitir a livre visualização e utilização por todos, o website foi alojado na plataforma *Heroku* e a base de dados em *GraphDB* foi alojada na cloud da *OntoText*.

Este website está disponível em <https://cryptonav.herokuapp.com/>.

4.1 Representação em grafos

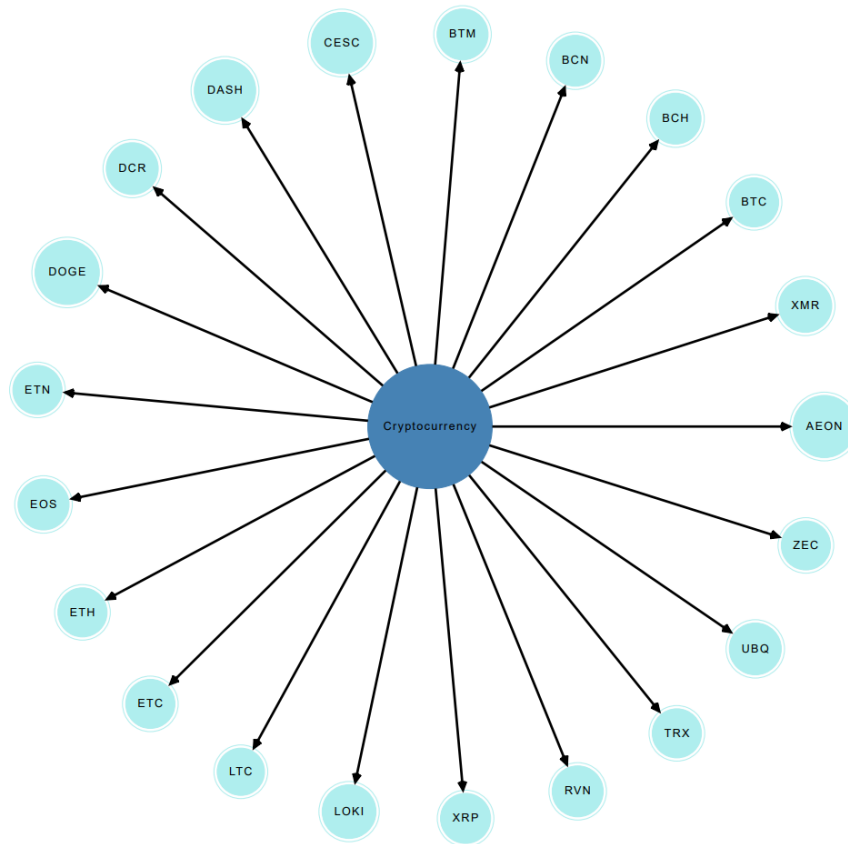


Figura 8: Grafo representativo de todas as *Cryptocurrency*.

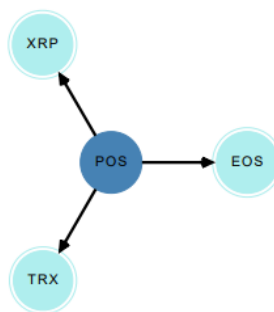


Figura 9: Grafo representativo de todas as *Cryptocurrency* baseadas em POS.

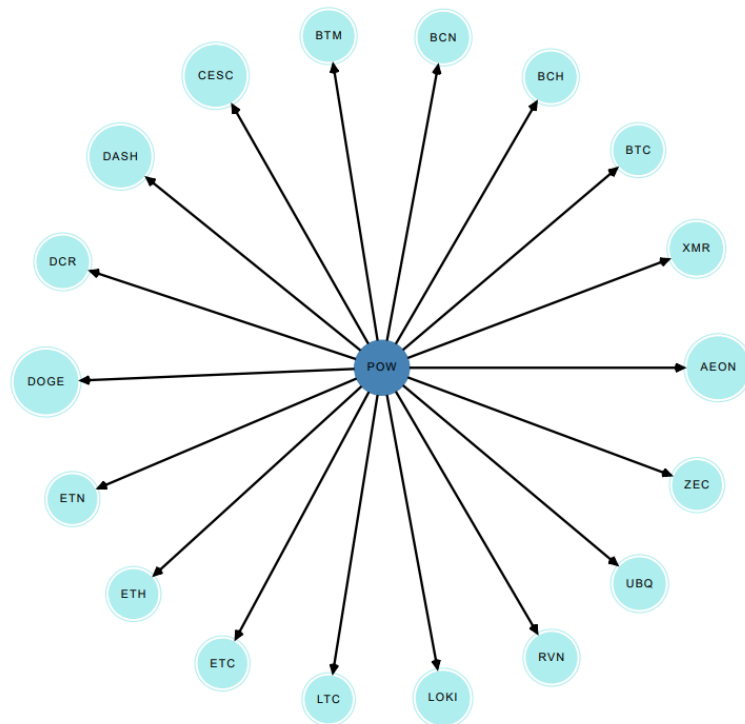


Figura 10: Grafo representativo de todas as *Cryptocurrency* baseadas em POW.

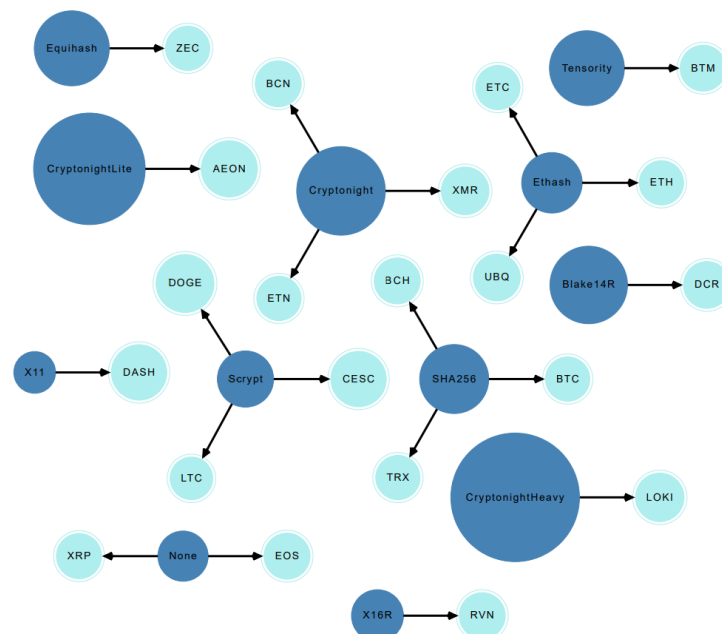


Figura 11: Grafo representativo de todos os algoritmos e as *Cryptocurrency* que os utilizam.

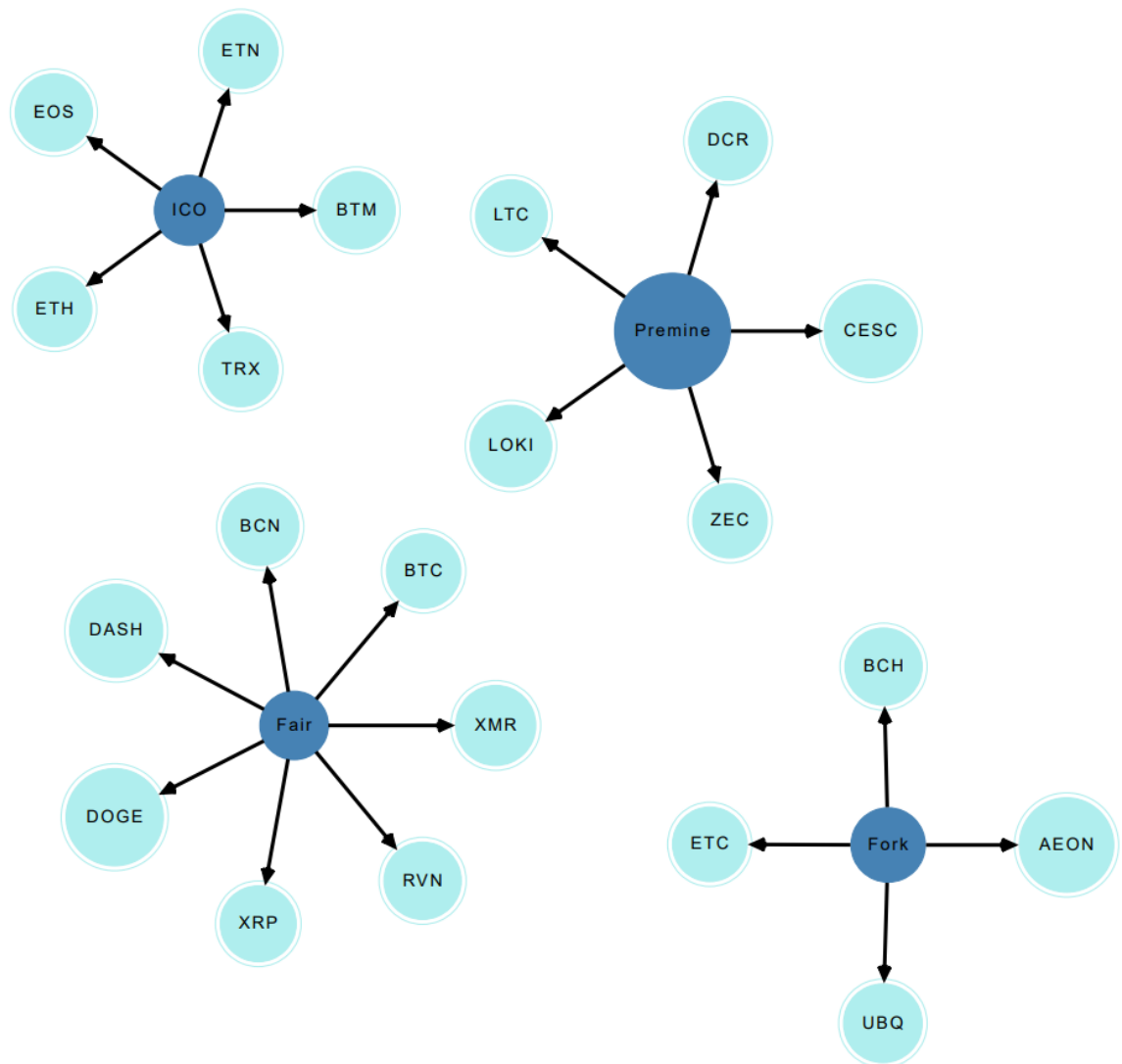


Figura 12: Grafo representativo da distribuição de todas as *Cryptocurrency*.

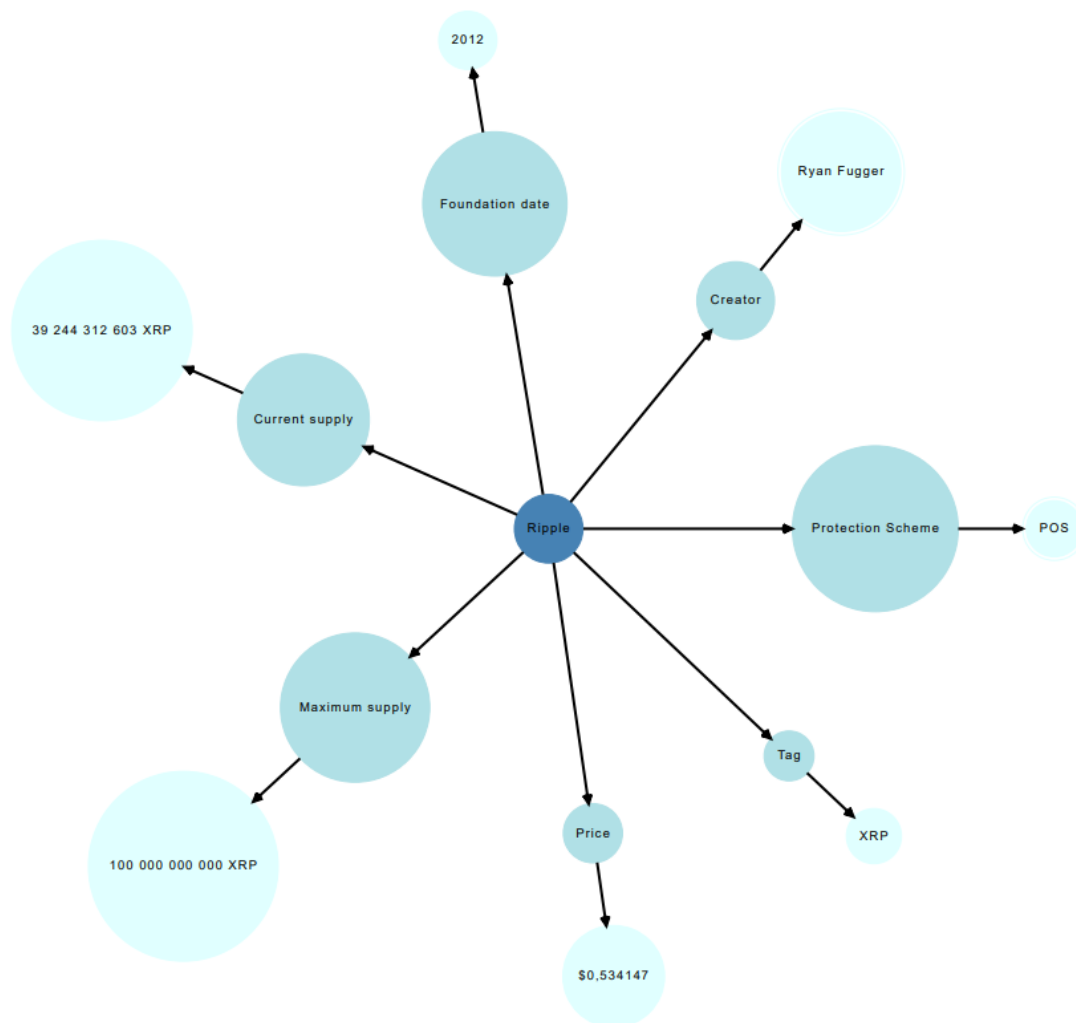


Figura 13: Grafo representativo de toda a informação presente numa *Cryptocurrency*. (neste exemplo foi utilizado a moeda *XRP* como exemplo, visto ser *POS* o que leva a ter menos informação para representar.)

4.2 Queries SPARQL

Nesta secção irão ser demonstradas algumas queries criadas em *SPARQL* que foram utilizadas no projeto.

4.2.1 Determinação de todas as *Cryptocurrency* e respetivas tags

```
select ?tag ?name where {  
  ?s a oacc:Cryptocurrency.  
  ?s oacc:tag ?tag.  
  ?s oacc:name ?name.  
}
```

4.2.2 Determinação dos criadores de todas as *Cryptocurrency*

```
select ?tag ?creator ?about ?photo where {  
  ?s a oacc:Cryptocurrency.  
  ?s oacc:tag ?tag.  
  ?s oacc:hasCreator ?c.  
  ?c oacc:name ?creator.  
  OPTIONAL{?c oacc:symbol ?photo.}  
  OPTIONAL{?c oacc:about ?about.}  
}
```

4.2.3 Determinação do algoritmo utilizado por cada *Cryptocurrency*

```
select ?coin_name ?alg_name where {  
  ?s a oacc:Cryptocurrency.  
  ?s oacc:name ?coin_name.  
  OPTIONAL{  
    ?s oacc:usingAlgorithm ?alg.  
    ?alg oacc:name ?alg_name.  
  }  
}
```

4.2.4 Determinação de toda a informação sobre todas as *Cryptocurrency*

```
select * where {  
  ?s a oacc:Cryptocurrency.  
  ?s oacc:name ?name.  
  OPTIONAL {?s oacc:isPOS ?pos.}  
  OPTIONAL {?s oacc:isPOW ?pow.}  
  OPTIONAL {?s oacc:about ?about.}  
  OPTIONAL {?s oacc:blockreward ?breward.}  
  OPTIONAL {?s oacc:blocktime ?btime.}  
  OPTIONAL {?s oacc:circulatingSupply ?csupply.}  
  OPTIONAL {?s oacc:founded ?founded.}  
  OPTIONAL {?s oacc:icoamount ?icoamount.}  
  OPTIONAL {?s oacc:maxsupply ?maxsupply.}  
  OPTIONAL {?s oacc:networkdif ?netdif.}  
  OPTIONAL {?s oacc:networkhashrate ?nethash.}  
  OPTIONAL {?s oacc:premineamount ?preamount.}  
  OPTIONAL {?s oacc:price ?price.}  
  OPTIONAL {?s oacc:projectwhitepaper ?whitepaper.}  
  OPTIONAL {?s oacc:symbol ?symbol.}  
  OPTIONAL {?s oacc:tag ?tag}  
  OPTIONAL {?s oacc:website ?website.}  
}
```

4.2.5 Determinação das *Cryptocurrency* e respetivo método de distribuição

```
select ?tag ?ico ?premine ?fair ?fork where {  
  ?s a oacc:Cryptocurrency.  
  ?s oacc:tag ?tag.  
  OPTIONAL{?s oacc:hasICO ?ico.}  
  OPTIONAL{?s oacc:hasPremine ?premine.}  
  OPTIONAL{?s oacc:hasNone ?fair.}  
  OPTIONAL{?s oacc:forkedFrom ?fork.}  
}
```

5 Conclusão

Neste projeto foi proposta a criação de uma ontologia para um domínio escolhido pelo aluno.

Primeiramente foi feito o levantamento de requisitos. Nesta fase foram tomadas decisões quanto aos possíveis conceitos a incluir e das funcionalidades a implementar.

De seguida fez-se a especificação do domínio que consistiu na criação de uma ontologia com os conceitos e os seus atributos.

O tema escolhido revelou-se extremamente interessante, e a implementação utilizando tecnologias lecionadas na aulas práticas revelou-se bastante acessível e prática.

Após a realização deste projeto, é possível dizer com confiança que este foi concluído com sucesso e de forma extremamente satisfatória.