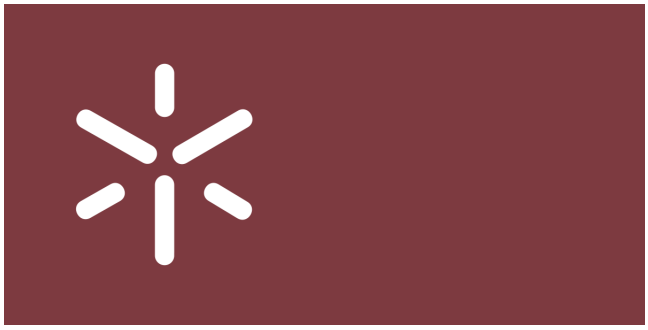

Universidade do Minho

Mestrado em Engenharia Informática

Processamento e Representação de Conhecimento

Cycling World

Trabalho Prático

Autor:
Frederico Pinto - 73639

14 de Junho de 2019

Conteúdo

1 Introdução **2**

2 Objetivos da Aplicação **2**

3 Domínio de Dados **2**

4 Ontologia **2**
4.1 Classes 2
4.2 Propriedades de Objetos 3
4.3 Propriedades de Dados 4

5 Aplicação **4**
5.1 Backend 4
5.2 Frontend 4

6 Conclusão **5**

1 Introdução

Este trabalho tem por objetivo obter informação e criar uma ontologia especificando o domínio dos dados. A partir dessa ontologia criar a base de dados e servir uma aplicação *web*, sendo essa dividida em *backend* e *frontend*. Neste relatório vou primeiro falar sobre o objetivo da aplicação, o domínio de dados envolvido e a prévia criação da ontologia.

Por fim irei falar sobre o desenvolvimento da aplicação *web*.

2 Objetivos da Aplicação

Esta aplicação tem como objetivo mostrar os resultados das provas oficiais de ciclismo de estrada realizadas. Para além disso, deve ser capaz de mostrar informação sobre todas as equipas profissionais registadas na *UCI* e seus empregados, atletas e staff.

3 Domínio de Dados

O domínio de dados envolvido para realizar o objetivo da aplicação é bastante grande e temos primeiro que pertencer o contexto em que estamos inseridos.

Quando falamos numa prova de ciclismo profissional, esta divide-se em duas categorias, prova de um dia ou prova de vários dias. Tendo na prova de um dia apenas uma classificação final, contudo nas provas de mais de um dia existem várias etapas que têm a sua própria classificação, para além disso existe uma classificação final que possui os resultados da prova para as várias camisolas referentes à prova.

Por outro lado, as equipas possuem os ciclistas e o staff que quiserem. Esses empregados possuem informações pessoais.

4 Ontologia

4.1 Classes

Após obtenção dos dados através de *datasets* e através da extração de informação de fontes oficiais, podemos agora começar a definir a nossa ontologia por base nessa informação que está inserida no dominio de dados e no objetivo da aplicação.

Numa primeira fase podemos definir várias classes como:

- Race
- Team
- Person

Como verificado na análise do dominio, uma corrida pode ser de dois tipos, sendo assim a classe *Race* possuí duas subclasses.

- Classic (prova de corrida)
- Tour (prova de várias etapas)

Para além disso a classe *Person* pode ser um atleta ou um membro do staff, logo surgem mais duas classes que são subclasses de *Person*.

- Athlete
- Staff

As corridas possuem etapas, logo surgiu a necessidade de criar a classe *Stage*.
Essas são as classes mais óbvias que a ontologia possuí, contudo surge agora um problema, esse problema consiste no fato de definir uma maneira de adicionar as classificações às corridas ou etapas. Então para o resolver, foi criada a classe *Classification*, essa classe possui como subclasse as seguintes classes.

- General
- Mountain
- Points
- Youth
- StageClassification

Sendo elas um tipo de classaficação existente mas diferente. Para além disso criei a classe *Position* que representa uma posição de uma classificação.

4.2 Propriedades de Objetos

Agora de maneira a relacionar as classes existentes foram criadas várias *object properties*.

- **hasTeam** (domain: Athlete, range: Team)
- **hasAthlete** (inverseof: hasTeam)
- **hasStaff** (domain: Team, range: Staff)
- **isStaffOf** (inverseof: hasStaff)
- **hasStage** (domain: Tour, range: Stage)
- **hasRace** (inverseof: hasStage)
- **hasClassification** (domain: Stage, range: StageClassification)
- **hasGeneral** (domain: Race, range: General)
- **hasMountain** (domain: Tour, range: Mountain)
- **hasPoints** (domain: Tour, range: Points)
- **hasYouth** (domain: Tour, range: Youth)
- **hasPosition** (domain: Classification, range: Position)

4.3 Propriedades de Dados

Relativamente a propriedades de dados, definimos várias conforme o que os *dataset* possuiam sendo assim apresentamos de seguida, juntamente com os domínios associados.

- **birthdate** (domain: Person)
- **class** (domain: Race)
- **continent** (domain: Team)
- **country** (domain: Team, Person, Race)
- **date** (domain: Stage, Race)
- **email** (domain: Team)
- **name** (domain: Race, Position, Person, Team, Stage)
- **rank** (domain: Position)
- **team** (domain: Position)
- **teamCategory** (domain: Team)
- **value** (domain: Position)
- **website** (domain: Team)

5 Aplicação

5.1 Backend

O *backend* foi criado utilizando o *Express*. Possui várias rotas para lidar com informação sobre provas, equipas, atletas e utilizadores. Essas rotas são *endpoints* para efetuar queries *Sparql* à base de dados em *GraphDB*.

Para além disso é implementada autenticação por *JSON-WebToken* que gera um *token* quando o *login*, devolvendo-o. Quando um pedido é recebido é recebido esse *token* tem que ser enviado de maneira a ser verificado para o pedido poder ser concretizado.

5.2 Frontend

Relativamente ao *frontend* utilizei o *Vue*. Aqui criei várias *views* que utilizam vários componentes para mostrar a informação sendo os *templates* construídos com ajuda do *Vuetify*.

Utilizou-se o roteador do Vue para direcionar as nossas rotas contudo para as protegermos, utilizamos o *Vuex* guardando um estado que é necessário existir para aceder às rotas exceto a rota de *login* e registo. Esse estado partilhado é o *token* necessário para efetuar os pedidos ao *backend*, que quando o *login* é efetuado é guardado através de uma mutação ou quando o *logout* é efetuado esse estado partilhado é colocado a *null*.

Tendo em vista a utilidade desta aplicação, o frontend foi ajustado para se adaptar a todo o tipo de ecrãs.

6 Conclusão

O objetivo da aplicação foi atingido com sucesso e o trabalho permitiu-me aprofundar os meus conhecimentos sobre ontologias e sobre a utilização do *Protégé*. Relativamente à parte da aplicação *web* permitiu alargar o conhecimento sobre *Vue*, trazendo o *Vuex* e *Vuetify*.