

UNIVERSIDADE DO MINHO  
MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA  
DEPARTAMENTO DE INFORMÁTICA

PROCESSAMENTO E REPRESENTAÇÃO DA  
CONHECIMENTO

---

# DotaTracker

---

Eduardo Gil Ribeiro da Rocha - **A77048**

13 de Junho de 2019

## **Resumo**

Este relatório tem como objetivo mostrar os passos tomados para desenvolver a plataforma *DotaTracker*, um *site* com o objetivo de permitir visualizar informação sobre o jogo *Dota 2*, abordando tanto o desenvolvimento do próprio *site* como os passos tomados para desenvolver a base de dados que o suporta.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Dota 2</b>	<b>3</b>
<b>3</b>	<b>Ontologia</b>	<b>3</b>
3.1	Match . . . . .	5
3.2	PlayerPerformance . . . . .	5
3.3	Player . . . . .	6
3.4	Hero . . . . .	6
3.5	Item . . . . .	6
<b>4</b>	<b>Estrutura da Aplicação</b>	<b>6</b>
<b>5</b>	<b>Conclusões</b>	<b>7</b>

# 1 Introdução

O primeiro objetivo deste trabalho é preencher uma base de dados com informação sobre partidas de *Dota 2*, tendo informação sobre cada partida, como por exemplo os jogadores participantes, personagens escolhidas, itens comprados, entre outros. O segundo objetivo é criar uma aplicação capaz de mostrar essa informação de forma conveniente ao utilizador.

Para além de abordar os processos usados para criar esta aplicação, este relatório irá descrever a ontologia criada para categorizar os dados armazenados.

Para criar a *front-end* do *site* foi usado *Vue.js* e *Vuetify*, enquanto que para a *back-end* foi usado *Express* e *GraphDB*.

## 2 Dota 2

*Dota 2* é um jogo de estratégia *multiplayer* em que as partidas se realizam entre duas equipas de cinco jogadores cada. A primeira equipa é chamada *Radiant* e a segunda *Dire*. Cada jogador escolhe uma personagem - ou Herói - e pode equipar essa personagem com até seis itens, que compra com ouro ganho durante o jogo.

O mapa está dividido em duas metades, cada metade tendo três *lanes*, que funcionam como as ruas do mapa. O objetivo do jogo é destruir a base da outra equipa, especificamente a estrutura no centro da base, o *Ancient*. A proteger o *Ancient* estão onze torres, duas na base e as restantes nove divididas pelas três *lanes* do mapa.

## 3 Ontologia

A informação usada no *site* é resultado de uma fusão de sete ficheiros, em total. Três destes ficheiros são do formato *CSV*, enquanto que os restantes são do formato *JSON*. Os primeiros três ficheiros são três ficheiros de amostra de uma base de dados com informação sobre jogos de *Dota 2*, com um tamanho total que ultrapassa os *500GB*. O primeiro ficheiro contém informação geral sobre os jogos, o segundo tem apenas registo do nível de perícia dos jogadores participantes e o terceiro tem mais informação específica de cada jogador.

Juntando estes três ficheiros, e limpando-os um pouco, formei a estrutura base sobre a qual este trabalho se construiu.

Um dos problemas que eu tive com os três ficheiros iniciais foi o fato de haver referências a *IDs* de heróis e itens, mas não informação sobre os próprios. Logo, procurei mais dois ficheiros, em formato *JSON*, do qual foram retirados os *IDs* usados nos ficheiros anteriores. Estes dois ficheiros tinham registo dos *IDs* dos heróis/itens, os seus nomes de código e os nomes localizados.

Para tornar a minha informação sobre itens e heróis um bocado mais completa, utilizei os dois últimos ficheiros, também em formato *JSON*, que para além de terem os seus próprios *IDs* e nomes de código iguais aos anteriores, tinham informação adicional, como descrições de itens e biografias de heróis. Então, realizei uma "fusão" de ficheiros, em que juntei as informações adicionais dos últimos ficheiros *JSON* à informação contida nos primeiros.

Antes de converter os ficheiros em formato *Turtle*, utilizei a aplicação *Protégé* para criar a base da minha ontologia. Finalmente, criei *scripts* capazes de transformar os ficheiros finais *CSV* e *JSON* num ficheiro de formato *Turtle*.

As classes da ontologia são as seguintes:

- Match - tem informação relativa a cada partida;
- PlayerPerformance - contém informação sobre a performance de um jogador em um jogo;
- Player - guarda informação de um jogador;
- Hero - tem os dados de uma personagem do jogo;
- Item - tem os dados de um item do jogo.

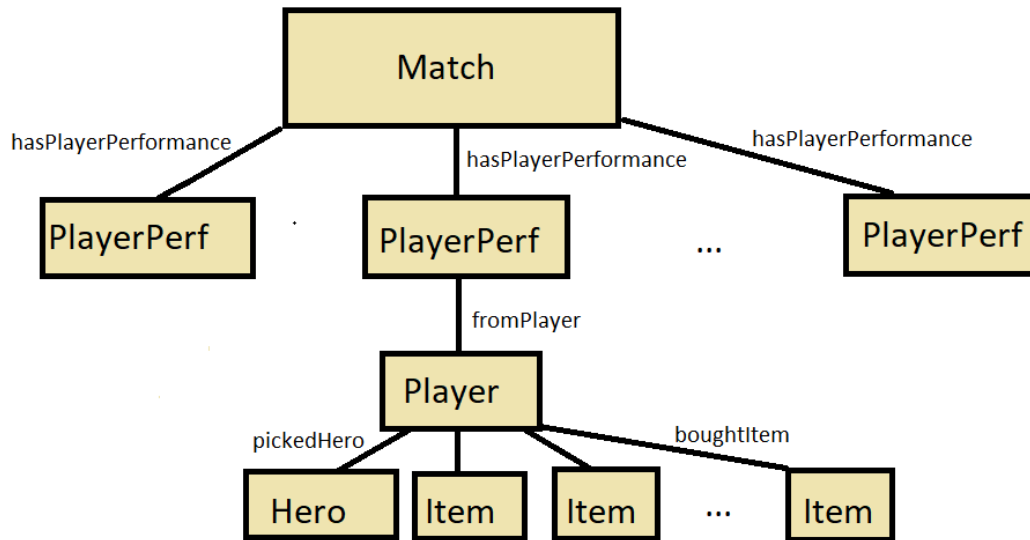


Figura 1: Representação da Estrutura da Ontologia

### 3.1 Match

A classe *Match* tem informação geral sobre cada jogo. Entre as relações de dados estão propriedades como a duração do jogo, tempo em que o jogo começou, nível de perícia dos jogadores, entre outros.

Um indivíduo da classe *Match* relaciona-se com dez indivíduos da classe *PlayerPerformance* - um por cada jogador.

### 3.2 PlayerPerformance

Esta classe guarda informação específica a cada jogador em cada jogo. As suas relações de dados encluem o número de vezes que em que matou outro jogador, o número de mortes, a quantidade de ouro ganha por minuto, a quantidade de experiência ganha por minuto, entre outros.

Existem ligações a três outras classes: uma à classe *Player* para registrar qual o jogador a que corresponde esta performance, uma à classe *Hero* para indicar qual foi a personagem escolhida e, finalmente, até seis ligações à classe *Item*, para se registrar quais itens é que o jogador comprou durante a partida.

### 3.3 Player

Esta é a classe mais pequena da ontologia, pois apenas guarda *IDs* de jogadores presentes em jogos. Não tem mais relações com outras classes para além da sua relação com as instâncias de *PlayerPerformance* que usam o jogador.

### 3.4 Hero

*Hero* é a classe que guarda as informações das personagens - ou heróis - do jogo. As suas relações de dados incluem *ID*, nome, nome localizado, biografia e listas de itens recomendados para esse herói.

### 3.5 Item

Por último, a classe *Item* é usada de forma semelhante à classe *Hero*, mas desta vez para os itens do jogo. Os itens, como os heróis, possuem também *ID*, nome e nome localizado. Para além desses três tipos de dados podem ter, opcionalmente, história do item, uma descrição do item e notas sobre uso do item.

## 4 Estrutura da Aplicação

Como referido anteriormente, foi usado *Vue.js* e *Vuetify* para a interface do programa e *Express* para o servidor que comunica com o *GraphDB*.

A aplicação possui seis tipos diferentes de páginas. O primeiro tipo é o índice da aplicação, onde se podem visualizar as partidas, numa tabela que pode ser ordenada pelos *IDs* dos jogos. As linhas da tabela expandem para revelar informação adicional sobre cada jogo. Carregar na parte expandida de uma partida deixa o utilizador entrar para o segundo tipo de página.

Este segundo tipo deixa o utilizador ver, para além da informação revelada no índice, as performances de todos os jogadores, incluindo imagens de todos os heróis e itens do jogo. Cada imagem serve como link para uma página diferente onde se pode ver informação específica a esse herói ou a esse item.

A terceira página é a lista de heróis, acessível através da *toolbar* no topo da aplicação. Esta página mostra apenas uma imagem de todos os heróis presentes na base de dados, sendo cada imagem também um *link*.

A quarta página é semelhante à terceira, visto que é também uma lista de imagens, mas desta vez de itens.

O quinto e sexto tipos de páginas são as páginas onde se veem informações específicas a um herói ou item, respetivamente.

## 5 Conclusões

Em suma, este relatório expôs a estrutura da ontologia, assim como as decisões tomadas durante a sua criação. Explicou também a estrutura do *site* e como navega-lo.

A realização deste trabalho permitiu aprofundar os meus conhecimentos em desenvolvimento de *front-ends*, assim como desenvolvimento de aplicações em geral, visto que a construção da aplicação precisou de mais do que *Vue.js* só.

A adição de informação aos ficheiros iniciou a partir dos quais eu criei a versão inicial da base de dados, apesar de ter aumentado o tempo de trabalho, acho que permitiu, no fim, ter uma aplicação mais completa, que se assemelha mais a um *site* que seria utilizado normalmente por um jogador de *Dota 2*.

Porém, apesar da informação acrescentada, acho que a aplicação beneficiaria também da existência de mais dados sobre os jogadores, como nomes ou registo de atividades adicionais, como uma lista de mensagens colocadas no *chat* do jogo. Mesmo apenas com os dados presentes, gostaria de ter posto uma página em que se pudesse visualizar as partidas em que cada jogador participou.