

INSTITUTO FEDERAL DE MINAS GERAIS  
CENTRO DE TECNOLOGIA  
PROGRAMA DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Guilherme Magno

**VISUALIZAÇÃO E PREDIÇÃO DO COMPORTAMENTO DE  
PARTIDA DO JOGO LEAGUE OF LEGENDS UTILIZANDO  
GRAFOS.**

BambuÍ, MG  
2018

**Guilherme Magno**

**VISUALIZAÇÃO E PREDIÇÃO DO COMPORTAMENTO DE PARTIDA DO JOGO  
LEAGUE OF LEGENDS UTILIZANDO GRAFOS.**

Trabalho de Conclusão de Curso apresentado  
ao Bacharelado em Engenharia de Computação  
do Instituto Federal de Minas Gerais (IFMG),  
como requisito parcial para a obtenção do grau  
de **Bacharel em Engenharia de Computação**

Orientador: Prof. Dr. Laerte Mateus

BambuÍ, MG

2018

Magno, Guilherme

Visualização e predição do comportamento de partida do jogo League of Legends utilizando grafos. / por Guilherme Magno. – 2018.  
32 f.: il.; 30 cm.

Orientador: Nome do orientador TEM Q CONFERIR ISSO  
Trabalho de Conclusão de Curso - Instituto Federal de Minas Gerais, ??????Centro de Tecnologia, Bacharelado em Engenharia de Computação, RS, 2018.

1. Modelo. 2. Latex. 3. Tcc. 4. Graduação. I. TEM Q CONFERIR ISSO, Nome do orientador. II. Visualização e predição do comportamento de partida do jogo League of Legends utilizando grafos. .

**Guilherme Magno**

**VISUALIZAÇÃO E PREDIÇÃO DO COMPORTAMENTO DE PARTIDA DO JOGO  
LEAGUE OF LEGENDS UTILIZANDO GRAFOS.**

Trabalho de Conclusão de Curso apresentado  
ao Bacharelado em Engenharia de Computação  
do Instituto Federal de Minas Gerais (IFMG),  
como requisito parcial para a obtenção do grau  
de **Bacharel em Engenharia de Computação**

**Aprovado em dia de mês de 2018:**

---

**Nome do orientador TEM Q CONFERIR ISSO, Dr. (UFSM)**  
(Presidente/Orientador)

---

**Nome membro banca Sobre nome, Me. (UFSM)**

---

**Nome membro banca Sobre nome, Tecg. (UFSM)**

BambuÍ, MG

2018

## **DEDICATÓRIA**

*Dedico este trabalho a .....*

## **AGRADECIMENTOS**

*Agradeço .....*

*“O valor de uma coisa depende da maneira como a abordamos mentalmente e não da coisa em si”*

(JIGORO KANO)

## **RESUMO**

### **VISUALIZAÇÃO E PREDIÇÃO DO COMPORTAMENTO DE PARTIDA DO JOGO LEAGUE OF LEGENDS UTILIZANDO GRAFOS.**

AUTOR: GUILHERME MAGNO

ORIENTADOR: NOME DO ORIENTADOR TEM Q CONFERIR ISSO

Aqui você escreve o resumo. Lembrando no máximo 250 palavras para tcc e 500 palavras para tese ou dissertação.

**Palavras-chave:** Modelo. latex. tcc. graduação.



# **ABSTRACT**

## **ABSTRACT TITLE**

**AUTHOR:** GUILHERME MAGNO

**ADVISOR:** NOME DO ORIENTADOR TEM Q CONFERIR ISSO

Here you write the summary. Remembering a maximum of 250 words for tcc and 500 words for thesis or dissertation.

**Keywords:** Model. latex. tcc. graduation.

## LISTA DE FIGURAS

Figura 1 –	Coeficiente de aglomeração .....	16
Figura 2 –	Mapa do jogo de <i>League of Legends</i> .....	17
Figura 3 –	As estruturas gráficas de uma batalha de MOBA .....	18
Figura 4 –	Representação visual de um objeto em JSON .....	20
Figura 5 –	Exemplo de retorno do uso da função <i>MATCH-V3</i> .....	21
Figura 6 –	Exemplo do uso do <i>WEBAPP</i> .....	25

## LISTA DE TABELAS

Tabela 1 –	Exemplo de <i>subset</i> salvo no banco de dados .....	23
Tabela 2 –	Exemplo da matriz gerada apos o processamento dos dados .....	24

## LISTA DE ABREVIATURAS E SIGLAS

MOBA	Arena de batalha online de multijogadores ( do inglês <i>Multiplayer Online Battle Arena</i> )
API	Interface de Programação de Aplicativos ( do inglês <i>Application Programming Interface</i> )
JSON	Notação de Objeto JavaScript ( do inglês <i>JavaScript Object Notation</i> )
URL	Localizador Padrão de Recursos ( do inglês <i>Uniform Resource Locator</i> )

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>13</b>
1.1	OBJETIVOS .....	13
1.1.1	<b>Objetivos Gerais.....</b>	<b>13</b>
1.1.2	<b>Objetivos Específicos e Resultados Esperados.....</b>	<b>13</b>
1.2	JUSTIFICATIVA .....	13
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>14</b>
2.1	GRAFOS .....	14
2.2	REDES COMPLEXAS .....	14
2.2.1	<i>Redes Small Worlds</i> .....	14
2.2.2	<b>Redes Livres de Escala .....</b>	<b>15</b>
2.2.3	<b>Redes Aleatórias.....</b>	<b>15</b>
2.2.4	<b>Coeficiente de Aglomeração .....</b>	<b>15</b>
2.3	<i>LEAGUE OF LEGENDS</i> .....	16
2.4	ESTADO DA ARTE .....	17
2.4.1	<b>Identifying Patterns in Combat that are Predictive of success .....</b>	<b>17</b>
2.4.2	<b>Análise de uma Métrica Alternativa para Predição de Laços Sociais em Grafos Lei de Potência.....</b>	<b>18</b>
<b>3</b>	<b>METODOLOGIA .....</b>	<b>19</b>
3.1	API DO <i>LEAGUE OF LEGENDS</i> .....	19
3.1.1	<b>JSON.....</b>	<b>19</b>
3.1.2	<b>MATCH-V3 .....</b>	<b>20</b>
3.2	BANCO DE DADOS .....	22
3.3	<i>PYTHON</i> .....	22
3.4	<i>WEBAPP</i> .....	24
<b>4</b>	<b>CONCLUSÃO .....</b>	<b>26</b>
	<b>REFERÊNCIAS .....</b>	<b>28</b>
	<b>APÊNDICES.....</b>	<b>29</b>
	<b>ANEXOS .....</b>	<b>31</b>

# 1 INTRODUÇÃO

Quando descrevemos uma situação utilizando pontos e algumas ligações entre esses pontos, estamos modelando este cenário em forma de grafos. E quando algo está modelado em forma de grafos,

## 1.1 OBJETIVOS

### 1.1.1 Objetivos Gerais

Criar um modelo computacional capaz de visualizar e prever a possibilidade de vitórias e derrotas no jogo *League of Legends* baseando-se num histórico de partidas usando redes complexas.

### 1.1.2 Objetivos Específicos e Resultados Esperados

1. Construir um *webservice* para visualização;
2. Calcular deterministicamente quantas vezes cada herói ganhou contra/com outro herói;
3. Testar o resultado obtido em outra amostra de dados;
4. Documentar os resultados.

## 1.2 JUSTIFICATIVA

Devido a crescente popularidade dos videogames no mundo e seus campeonatos mundiais valendo milhões de dólares, a criação deste modelo computacional capaz de visualizar e prever a possibilidade de vitórias e derrotas, [continua...]

## 2 FUNDAMENTAÇÃO TEÓRICA

Nessa parte serão apresentados conceitos para melhor entendimento deste trabalho e da modelação utilizada. A seção 2.1 elucida o que são grafos e o capítulo 2.2 explica a variação de grafos utilizado para modelar o problema proposto. Depois, na seção 2.3 é esclarecido o ambiente que será abstraído, explicando algumas regras básicas do jogo.

### 2.1 GRAFOS

Segundo VIANA (2007) "Um grafo  $G(V, E)$  pode ser definido como um conjunto de vértices  $V$ , e um conjunto de conexões  $E$ " e ele continua: "Cada elemento do conjunto  $E$ , associa dois elementos do conjunto  $V$ , assim, se  $(u, v) \in E$ , então existe uma conexão entre os vértices  $u$  e o vértice  $v$ ."

Os grafos também podem ser classificados como binários e não-binários, sendo os binários suas arestas representadas por 1 ( se existem ) ou 0 ( se não existem ), e os não-binários quando existe um conjunto  $W$ , onde informa a intensidade da interação de uma aresta entre dois vértices (VIANA, 2007). (continua...)

### 2.2 REDES COMPLEXAS

Sobre redes complexas, VIANA (2007) diz que é utilizado o termo redes complexas, quando um grafo representa um sistema físico real. Então, levando em conta isso, um grafo do jogo LOL pode ser considerado como uma rede complexa.

(continua...)

Para que seja possível classificar os resultados adequadamente, serão apresentados três modelos que se destacam no estado da arte, as redes *small worlds*, as redes livres de escala (ALBERT; BARABÁSI, 2002), e as redes aleatórias. E também será explicado sobre o coeficiente de aglomeração .

#### 2.2.1 Redes Small Worlds

As redes *small worlds*, são redes que, o caminho entre dois nós são relativamente pequenos. (continua...).

### 2.2.2 Redes Livres de Escala

Nas redes livres de escala, um nó tem a probabilidade  $P(k)$  de possuir  $k$  arestas obedecendo a lei da potência  $P(k) \sim k^{-\gamma}$  (ALBERT; BARABÁSI, 2002), (ANTIQUERA et al., 2005). Segundo Matheus Palhares Viana, nas redes livres de escala, muitos nós tem poucas arestas, e poucos nós se ligam a muitos.

### 2.2.3 Redes Aleatórias

Segundo VIANA (2007), as redes aleatórias são um sistema formado por  $E$  arestas e  $N$  vértices, onde as arestas são distribuídas aleatoriamente. CUNHA RECUERO (2004) exemplifica esse tipo de rede como uma festa, “onde bastaria uma conexão entre cada um dos convidados, para que no fim dela, todos estejam conectados no fim dela” e que quanto mais conexões forem criadas, maior a chance de serem criados grupos de pessoas, e que de tempos se relacionarem com outros grupos, e que poderia concluir que esses nós se relacionarem de forma randômica.

### 2.2.4 Coeficiente de Aglomeração

Segundo VIANA (2007), o coeficiente de aglomeração mede o quão conectado estão os nós da rede ou do grafo. ANTIQUERA et al. (2005) define o coeficiente de aglomeração sendo

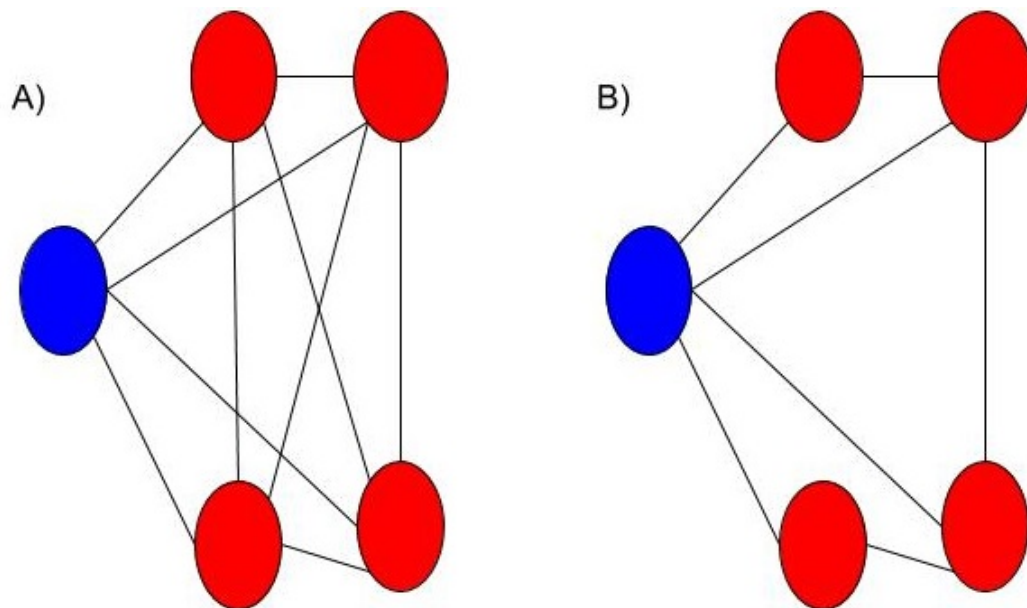
$$CA_i = \frac{E_i}{k_i(k_i - 1)}$$

ANTIQUERA et al. (2005) continua: “Sendo para cada vértice  $i$  existe  $k_i$  arestas, que os ligam a outros  $k_i$  vértices. Se esses  $k_i$  vértices estivessem ligados diretamente à todos os outros vértices do conjunto, haveriam  $k_i(k_i - 1)$  arestas entre eles. E assumindo  $E_i$  o número de arestas que existentes entre os  $k_i$  vértices. O coeficiente da rede inteira é a média de todos  $CA_i$ ”

No grafo A da Figura 1, assumindo o peso de todas as arestas como 1, o coeficiente de aglomeração do vértice em azul é 1, e no grafo B o coeficiente do vértice em azul, segundo a fórmula é 12.



Figura 1 – Coeficiente de aglomeração



Fonte: Autor (2018).

### 2.3 LEAGUE OF LEGENDS

O jogo *League of Legends*, é um jogo classificado como arena de batalha online de multi jogadores (do inglês *Multiplayer Online Battle Arena* ) ou conhecido também como MOBA, que é um estilo de jogo onde, duas equipes se enfrentam em um campo de batalha, e cada jogador controla o seu personagem, mais chamado de herói ou campeão. O objetivo do MOBA é derrotar a equipe adversária destruindo a construção principal da equipe inimiga.

A arena onde acontece o jogo é uma arena onde normalmente o mapa inicialmente é espelhado, ou seja, o lado que cada time está, não oferece vantagens exclusivas. O mapa é composto de três caminhos até a base inimiga.

A Figura 2 mostra como é realmente o mapa do jogo, sendo que os círculos mostram a localização da construção principal, e os triângulos as construções de suporte de cada equipe, sendo azul uma equipe, e vermelho a outra.

Com o início do jogo, cada jogador escolhe um herói diferente, onde cada herói tem um conjunto de características únicas, como habilidades especiais, seu impacto no jogo e nas equipes adversárias e aliadas.

Depois de escolher os heróis de cada equipe, cada jogador, deve procurar adquirir recursos no jogo, e objetivos para conseguir vantagens. Os recursos são limitados por equipe e por

Figura 2 – Mapa do jogo de *League of Legends*



Fonte: Autor (2018).

tempo, ou seja, deve ser bem escolhido quem ficará com a maior parte dos recursos da equipe.

## 2.4 ESTADO DA ARTE

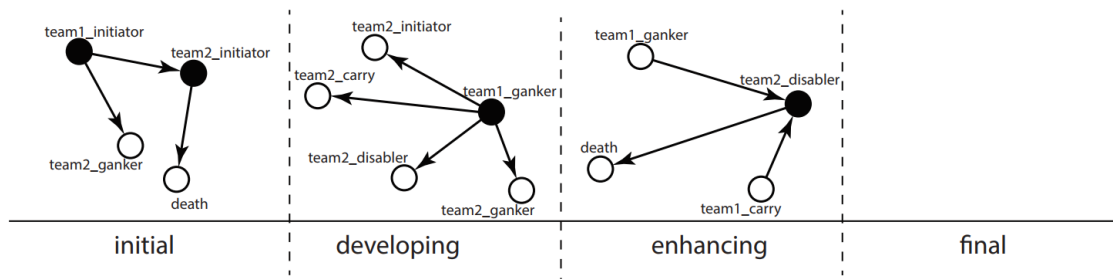
No estado da arte, estudos com jogos de gênero MOBA são poucos, mas relacionados a redes complexas e predição existem diversos.

### 2.4.1 Identifying Patterns in Combat that are Predictive of success

O trabalho de Yang *et al*, relacionou a posição dos jogadores em uma batalha do MOBA com redes complexas, ao predizer o time vencedor daquela batalha.

Yang *et al* modela os dados de um combate, em redes complexas, e treina uma árvore

Figura 3 – As estruturas gráficas de uma batalha de MOBA



Fonte: (YANG; HARRISON; ROBERTS, 2014).

de decisão usando as melhores características tiradas dessa rede complexa. Depois de treinado a árvore, é identificado regras de combate que são preditivas de sucesso, e depois essas regras são traduzidas de volta em um padrão de combate específico usando uma técnica chamada mineração de subgrafo frequente.

#### 2.4.2 Análise de uma Métrica Alternativa para Predição de Laços Sociais em Grafos Lei de Potência

Feito por DANIELEWICZ, ela mostra uma visão geral do problema de predição de laços sociais, analisa os modelos de geração de grafos principalmente que seguem a lei da potência, no âmbito de formação de laços sociais e depois consegue uma métrica para predição de laços. Ela faz testes usando técnicas diferentes, mostrando seus resultados.

### 3 METODOLOGIA

Para a realização do projeto, é necessário antes explicar as tecnologias usadas, e o porquê foram usadas. A primeira seção irá explicar como os dados estão disponíveis e de que informações eles carregam. A segunda fala sobre a linguagem de programação utilizada para conseguir os dados, verificar se os dados estão completos, e armazená-los em um banco de dados. A terceira clarifica o que é um banco de dados, e porque foi utilizado.

#### 3.1 API DO *LEAGUE OF LEGENDS*

A *Riot Games*, desenvolvedora e dono do jogo *League of Legends*, fornece uma Interface de Programação de Aplicativos ( do inglês *Application Programming Interface* ) ou chamado de API para que terceiros consigam acessar dados sobre os jogos.

Para conseguir ter acesso à essa API, é necessário se cadastrar, entrar no *site* na parte de desenvolvedores, disponível em <http://developer.riotgames.com> ( não sei se pode por link assim ). Depois de cadastrado e de se credenciar no *site*, deve ser clicado no botão para gerar uma chave API, que é válida por apenas alguns dias, a menos que cadastre o projeto, dessa maneira a chave é válida por um ano.

O acesso à essa API é por URL utilizando uma função disponível pela *RIOT Games*. Neste trabalho, usaremos apenas a função *MATCH-V3*, que é uma função que retorna os dados de uma partida já terminada. Nas próximas subseções serão abordados o formato no qual a API disponibiliza os dados, e quais informações ela retorna com o uso do *MATCH-V3* e quais serão usados no projeto.

##### 3.1.1 JSON

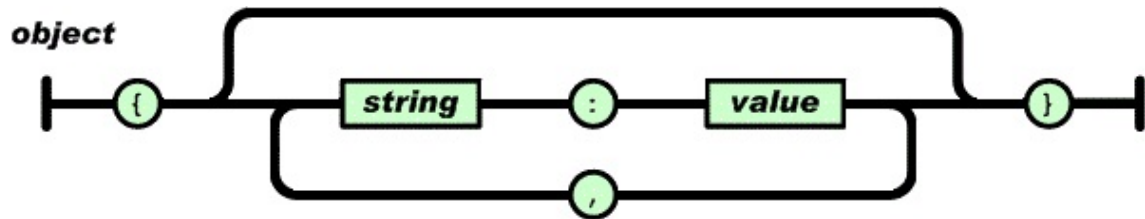
(preciso achar outros autores para respaldarem)

A API do *League of Legends* disponibiliza as informações no formato de Notação de Objeto JavaScript ( do inglês *JavaScript Object Notation*. ) ou JSON. O JSON é um formato de dados que é escrito em forma de texto. O padrão JSON é um jeito de escrever as informações casando o nome das variáveis com um valor, ou em uma lista ordenada de valores. E o objeto em JSON é uma lista desordenada de valores (JSON, 1999).

Segundo o JSON "JSON [...] é uma formatação leve de troca de dados. Para ser

humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar."e continua explicando "JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares [...]. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados."

Figura 4 – Representação visual de um objeto em JSON



Fonte: (JSON, 1999).

### 3.1.2 MATCH-V3

A função *MATCH-V3* é disponibilizada publicamente pela *RIOT Games* e que retorna um conjunto de informações no formato JSON sobre a partida passada por parâmetro, quando essa partida existe no banco de dados da *RIOT Games*. A Figura 5 resume um exemplo de acesso utilizando a função acessando a URL <https://br1.api.riotgames.com/lol/match/v3/matches/1381102minhachave> acessada em 21 de março de 2018, sendo que *minhachave* tem que ser substituída por uma chave privada válida.

De todas essas informações retornadas pela função, as informações armazenadas para o projeto por participante são:

 *gameId*. Identificador único da partida;

- *kills, deaths e assists*. Informações que falam, respectivamente, quantas vezes esse jogador matou, morreu e participou na morte de outrem;

- *win*. Se ele ganhou;

- *championId*. Qual campeão ele escolheu;

- *lane*. Em qual posição ele tava jogando;

- *platformId*. Em qual servidor ele jogava;

Figura 5 – Exemplo de retorno do uso da função *MATCH-V3*

```

"gameId":1381102031,
"platformId":"BR1",
"gameCreation":1526653171409,
"gameDuration":1533,
"queueId":420,
"mapId":11,
"seasonId":11,
"gameVersion":"8.10.229.7328",
"gameMode":"CLASSIC",
"gameType":"MATCHED_GAME",
"teams":[
  {
    "teamId":100,
    "win":"Win",
    "firstBlood":true,
    "firstTower":true,
    "firstInhibitor":true,
    "firstBaron":true,
    "firstDragon":true,
    "firstRiftHerald":true,
    "towerKills":10,
    "inhibitorKills":2,
    "baronKills":1,
    "dragonKills":2,
    "vilemawKills":0,
    "riftHeraldKills":1,
    "dominionVictoryScore":0,
    "bans":[
  ],
  {
  },
],
"participants":[
  {
    "participantId":1,
    "teamId":100,
    "championId":420,
    "spell1Id":12,
    "spell2Id":4,
    "highestAchievedSeasonTier":"PLATINUM",
    "stats":{
    },
    "timeline":{
    },
  },
  {
  },

```

Fonte: Autor.

- *queueId*. E qual o tipo de partida.

Sabendo qual informação vamos armazenar, ficou escolhido a linguagem *Python* para ser desenvolvido o algoritmo no qual obterá as informações de forma não manual e o sistema de gerenciamento de banco de dados MySQL. Na próxima seção será apresentado sobre banco de dados e na seção 3.3 será esclarecido o que é linguagem de programação *Python* e como foi usado.

## 3.2 BANCO DE DADOS

(continua)

## 3.3 PYTHON

*Python* é uma linguagem de programação dentre muitas, assim como existem várias linguagens faladas e escritas por nós humanos, no mundo da computação, existem várias linguagens de programação, onde geralmente cada uma se destaca em pelo menos um quesito.

(pegar algum autor para respaldar, e pegar algum texto dele falando sobre vantagens do *Python*, que é interpretado e continua.)

Entendendo isso, foi criado um algoritmo para a aquisição dos dados de partidas pela API do *League of Legends*, que usava a biblioteca *requests* do *Python* para conseguir acessar a API por URL's, e a biblioteca *json* também do *Python* para transformar o Json retornado pela requisição da URL em um objeto em *Python*.

O aplicativo verifica se a partida requisitada existe e se os dados necessários estão completo, salvando no banco de dados apenas as partidas válidas. O pseudo-código se encontra no Algoritmo 1.



---

**Algoritmo 1: AQUISIÇÃO DOS DADOS DAS PARTIDAS**


---

```

início
  i = 1000000000; /* Esta é a id de uma partida aleatória na ultima versão do
  jogo*/
  chave = minha chave privada de acesso;
  se Existe arquivo com ultimo partida lido então
    | i = ultima partida lida;
  repita
    pedido = requisita URL Da API(i, chave);
    se pedido.status == 200 então
      /* se a partida existe */ JSON = carrega JSON Do Pedido(pedido);
      se JSON é válido então
        | Armazena os dados em um banco de dados;
        i = i + 1;
        Salva i no arquivo;
    até Até ser pausado;
fim

```

---

Onde com esse algoritmo foram armazenadas 28597 jogos válidos diferentes, e foi decidido usar apenas as partidas ranqueadas 5 contra 5, já que estas são as partidas responsáveis pela classificação dos jogadores dentro dos jogo, e as partidas não ranqueadas são consideradas como amistosas. Com esse escopo, o número de partidas usados foram ao todo 11717 partidas ranqueadas diferentes, na tabela 2 é possível ver uma pequena fatia dos dados salvos.

**Tabela 1 – Exemplo de *subset* salvo no banco de dados**

match_id	kills	deaths	assists	win	champ_id	lane	player_id	platform	type
1282000002	11	10	7	0	67	BOTTOM	16112724	BR1	420
1282000002	2	8	15	0	412	BOTTOM	18604874	BR1	420
1282000002	7	4	7	0	34	MIDDLE	19281809	BR1	420
1282000002	7	6	4	0	5	JUNGLE	21304223	BR1	420
1282000002	5	7	12	0	98	TOP	651014	BR1	420
1282000002	0	6	23	1	44	BOTTOM	18592234	BR1	420
1282000002	19	5	5	1	222	BOTTOM	7170345	BR1	420

Fonte: Autor.

Com esse número de partidas, foi feito um outro *software*, para processamento dos dados, onde é contabilizado quantas vezes cada campeão jogou com outro campeão, seja no mesmo time ou no time adversário, calculado quantas vitória contra, e com o outro herói. Na tabela 2 é possível ver um exemplo de como a estrutura fica apos o processamento dos dados, sendo *id*, o identificador de cada campeão, e *vj*, *vc*, *nj* e *nc* sendo respectivamente as vitórias juntos, vitórias contra, número de jogos juntos, números de jogos contra.



 Tabela 2 – Exemplo da matriz gerada apos o processamento dos dados

id x id		1				2				3			
		vj	vc	nj	nc	vj	vc	nj	nc	vj	vc	nj	nc
1	vj												
	vc					7	3	13	11	11	23	17	33
	nj												
	nc												
2	vj		7										
	vc		8							5	8	9	11
	nj		13										
	nc		11										
3	vj		11				5						
	vc		10				3						
	nj		17				9						
	nc		33				11						

Fonte: Autor.

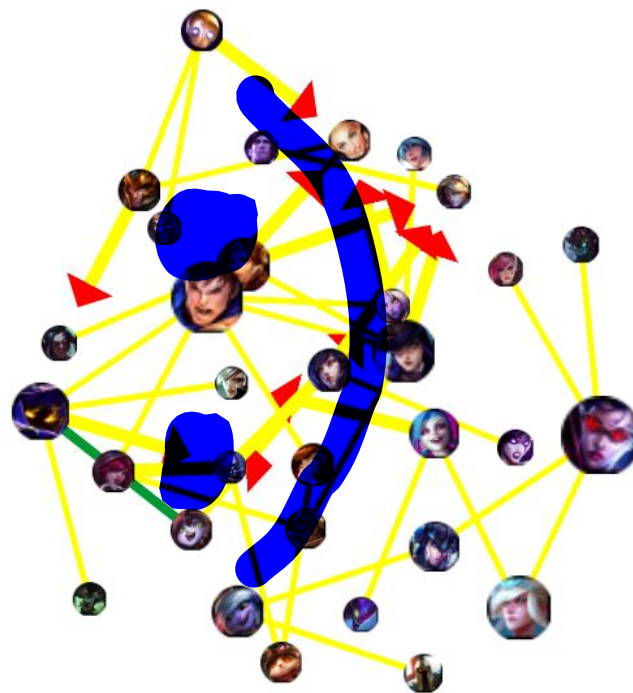
E com as informações já processados, foi montado um *webapp*, que sera explicado na proxima seção, para uma melhor visualização dos dados obtidos.

### 3.4 WEBAPP

(continua...)

Com essa aplicação, é possível visualizar o resultado de todos os dados no banco de dados ou diminuir o escopo, filtrando alguns dados. Na figura 6 é possível ver um exemplo

Figura 6 – Exemplo do uso do *WEBAPP*



Fonte: Autor.

## **4 CONCLUSÃO**

Está é a conclusão do trabalho ....



## REFERÊNCIAS

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. **Reviews of modern physics**, [S.l.], v.74, n.1, p.47, 2002.

ANTIQUEIRA, L. et al. Modelando textos como redes complexas. In: III WORKSHOP EM TECNOLOGIA DA INFORMAÇÃO E DA LINGUAGEM HUMANA. **Anais...** [S.l.: s.n.], 2005. p.22–26.

CUNHA RECUERO, R. da. Teoria das redes e redes sociais na internet: considerações sobre o orkut, os weblogs e os fotologs. In: XXVII CONGRESSO BRASILEIRO DE CIÊNCIAS DA COMUNICAÇÃO. XXVII INTERCOM. **Anais...** [S.l.: s.n.], 2004.

DANIELEWICZ, G. Análise de uma métrica alternativa para predição de laços sociais em grafos lei de potência. ?, [S.l.], 2016.

JSON. **Introdução ao JSON**. 1999.

VIANA, M. P. **Metodologia das redes complexas para caracterização do sistema de Havers**. 2007. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.

YANG, P.; HARRISON, B. E.; ROBERTS, D. L. Identifying patterns in combat that are predictive of success in MOBA games. In: FDG. **Anais...** [S.l.: s.n.], 2014.

## APÊNDICES

---

## .1 TESTE

# ANEXOS

---



**ANEXO A – Título do Anexo**

Este é o anexo A