



INE5454 - Trabalho Web Scraping

César Augusto Pereira de Souza



Sumário

1. Introdução e Objetivo
2. O site vlr.gg
3. Estruturação do Projeto
4. Análise dos Resultados Finais
5. Considerações Finais



1

Introdução e Objetivo

Definições

- *Valorant*: jogo da empresa Riot Games, que possui um cenário competitivo dinâmico, com torneios globais de alto nível e uma crescente comunidade de jogadores profissionais e fãs. O jogo é um dos mais jogados no mundo, e sempre destaca-se pela sua vasta audiência nos torneios, chegando a atingir mais de 1.2 milhão de telespectadores simultâneos, somando todas as plataformas, no campeonato mundial de 2023.
- *vlg.gg*: principal portal de notícias e informações sobre o cenário competitivo de *Valorant*. Possui rankings para todas as regiões com cenário competitivo ativo no mundo, listando estatísticas de desempenho de quase todos os times possíveis.

Objetivo do trabalho

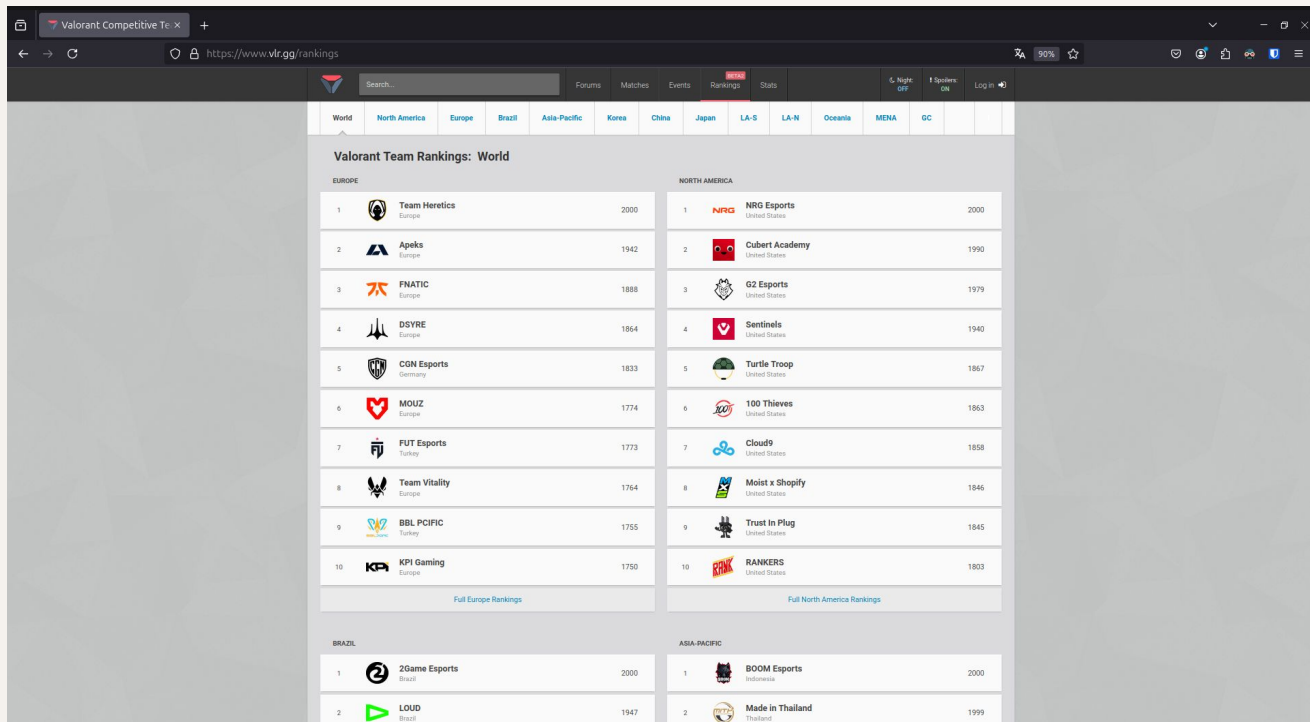
- O *Valorant* competitivo também possui um mercado em casas de aposta agitado. Informações acerca dos times, principalmente dados estatísticos, podem ser úteis, tanto para que um time identifique pontos de desvantagem, que afetam no seu desempenho, quanto para cenários de simulação, permitindo o cálculo de probabilidades de vitória de cada time, número de que serão rounds jogados, etc.
- Desse modo, o objetivo deste trabalho é construir um Web Scraper, baseado em um extrator de dados, para obter informações sobre todos os times listados nos rankings regionais. O scraper deve “receber” como entrada a página de rankings regionais do *vlr.gg*, e irá retornar o *dataset* coletado em um arquivo JSON.
- Chamaremos a ferramenta de **VLRGG Scraper**.



























2

0 site vlr.gg

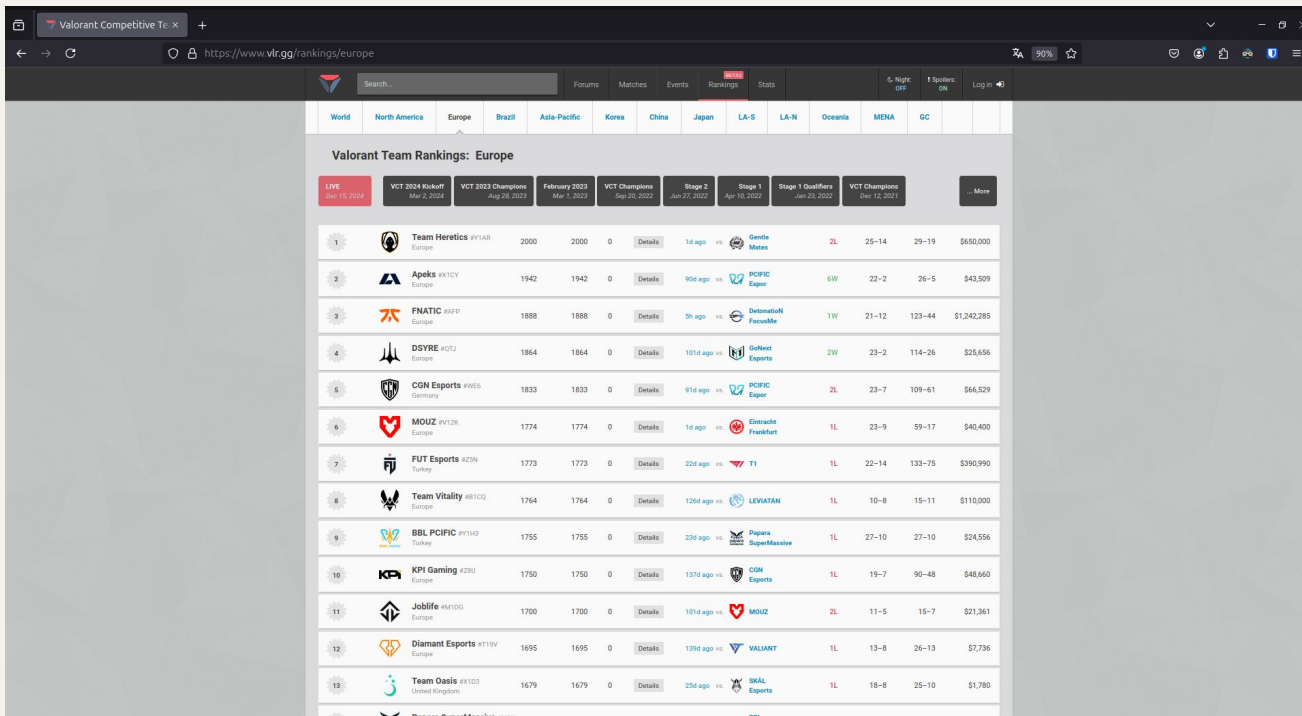
Rankings Regionais





























The screenshot shows the Valorant Competitive website with the 'Rankings' tab selected. The page displays 'Valorant Team Rankings: World' and lists teams across four regions: Europe, North America, Brazil, and Asia-Pacific. Each team entry includes a rank, logo, team name, region, and MMR score. Navigation links for each region are provided at the bottom of each list.

World	North America	Europe	Brazil	Asia-Pacific	Korea	China	Japan	LA-S	LA-N	Oceania	MENA	CC
Valorant Team Rankings: World												
EUROPE												
1		Team Heretics	Europe	2000								
2		Apeks	Europe	1942								
3		FNATIC	Europe	1888								
4		DSYRE	Europe	1864								
5		CGN Esports	Germany	1833								
6		MOUZ	Europe	1774								
7		FUT Esports	Turkey	1773								
8		Team Vitality	Europe	1764								
9		BBL PCIFIC	Turkey	1755								
10		KPI Gaming	Europe	1750								
Full Europe Rankings												
NORTH AMERICA												
1		NRG Esports	United States	2000								
2		Cubert Academy	United States	1990								
3		G2 Esports	United States	1979								
4		Sentinels	United States	1940								
5		Turtle Troop	United States	1867								
6		100 Thieves	United States	1863								
7		Cloud9	United States	1858								
8		Moist x Shopify	United States	1846								
9		Trust In Plug	United States	1845								
10		RANKERS	United States	1803								
Full North America Rankings												
BRAZIL												
1		2Game Esports	Brazil	2000								
2		LOUD	Brazil	1947								
ASIA-PACIFIC												
1		BOOM Esports	Indonesia	2000								
2		Made in Thailand	Thailand	1999								

Ranking Regional seleccionado (Europe)




The screenshot shows the Valorant Competitive website with the 'Rankings' tab selected. The 'Europe' regional ranking is displayed, listing 13 teams. Each team entry includes its rank, logo, name, region, MMR, and a record of recent matches with their opponents and scores. The website interface includes a search bar, navigation tabs for various regions, and a top navigation bar with links to Forums, Matches, Events, and Rankings.

World	North America	Europe	Brazil	Asia-Pacific	Korea	China	Japan	LA-S	LA-N	Oceania	MENA	CC
Valorant Team Rankings: Europe												
LIVE Dec 15, 2024	VCT 2024 Kickoff Mar 2, 2024	VCT 2023 Champions Aug 26, 2023	February 2023 Mar 1, 2023	VCT Champions Sep 20, 2022	Stage 2 Jun 27, 2022	Stage 1 Apr 16, 2022	Stage 1 Qualifiers Jan 23, 2022	VCT Champions Dec 12, 2021	... More			
1		Team Heretics #V1AR Europe	2000	2000	0	Details	1d ago vs  Gen.G Match	2L	25-14	29-19	\$650,000	
2		Apeks #V1CY Europe	1942	1942	0	Details	9d ago vs  PCFIC Esper	6W	22-2	26-5	\$43,509	
3		FNATIC #V1FP Europe	1888	1888	0	Details	5h ago vs  Detonation FancatMe	1W	21-12	123-44	\$1,242,285	
4		DSVRE #V1GJ Europe	1864	1864	0	Details	101d ago vs  Gulfport Esports	2W	23-2	114-26	\$25,656	
5		CGN Esports #V1ES Germany	1833	1833	0	Details	91d ago vs  PCFIC Esper	2L	23-7	109-61	\$66,529	
6		MOUZ #V1DK Europe	1774	1774	0	Details	1d ago vs  Eintracht Frankfurt	1L	23-9	59-17	\$40,400	
7		FUT Esports #V1CN Turkey	1773	1773	0	Details	22d ago vs  T1	1L	22-14	133-75	\$390,990	
8		Team Vitality #V1CQ Europe	1764	1764	0	Details	126d ago vs  LEVATIAN	1L	10-8	15-11	\$110,000	
9		BBL PCFIC #V1H3 Turkey	1755	1755	0	Details	23d ago vs  Papara SuperMassive	1L	27-10	27-10	\$24,556	
10		KPI Gaming #V1SU Europe	1750	1750	0	Details	137d ago vs  CGN Esports	1L	19-7	90-48	\$48,660	
11		Joblife #V1DG Europe	1700	1700	0	Details	101d ago vs  MOUZ	2L	11-5	15-7	\$21,361	
12		Diamant Esports #V1VW Europe	1695	1695	0	Details	129d ago vs  VALMANT	1L	13-8	26-13	\$7,736	
13		Team Oasis #V1D3 United Kingdom	1679	1679	0	Details	25d ago vs  SKAL Esports	1L	18-8	25-10	\$1,780	
... More												

Visão Geral de um time

Team Heretics: Valorant

https://www.vlr.gg/team/1001/team-heretics



Team Heretics TH
teamheretics.com
@hereticsthal
EUROPE

Overview


Stats

Matches

News

Transactions

UPCOMING MATCHES

 VCT 2025 EMEA Kickoff
USQF


Team Heretics #1 LAR

TBD

TBD

2025/01/16
6:00 pm

RECENT RESULTS


 HereticsXP #2 Showmatch - Main Event

Team Heretics #1 LAR

0 : 2

Gentle Mates #2 DRI

2024/12/13
3:30 pm


 Red Bull HGS #5 R3

Team Heretics #1 LAR

0 : 1

FUT Esports

2024/11/22
10:40 am


 Red Bull HGS #5 R2

Team Heretics #1 LAR

0 : 1

Cloud9 #1 CF

2024/11/21
2:10 pm


 Red Bull HGS #5 R1

Team Heretics #1 LAR

1 : 0

FOKUS #1 LLE

2024/11/21
10:15 am

 VCT 2024 Champions Playoffs - GF

Team Heretics #1 LAR

2 : 3

Edward Gaming #1 CJ

2024/08/25
4:00 am

RELATED NEWS

2024/11/21
Team Heretics and Cloud9 victorious to open Red Bull Home Ground

2024/11/20
benjyfishy: "We never put expectations on ourselves"

2024/08/25
Edward Gaming conquer Champions 2024 in historic win for China


2024/08/24
Team Heretics topple Leviatan, advance to Champions 2024 grand final


2024/08/23
Team Heretics beat Sentinels, will face Leviatan in lower finals


34 more articles


CURRENT ROSTER


PLAYERS

 Boo #1
Konstantin Lukashenko


 MiroBoo
Dmitry Lukashenko


 VioZ
Mert Akman


 kicHo
Ezek Esenli

 benjyfishy
Benjamin David Fish


START


 Niklas
Niklas Gettes
MANAGER

 nelzinho
Neil Frey
HEAD COACH

 weber
Brandon Weber
COACH

EXPLORE

 ExpertoRyan

 Pablo

EVENT PLACEMENTS

TOTAL WINNINGS

\$779,413

Red Bull Home Ground #5

2024

Main Event - \$50-60k

Valorant Champions 2024

2024

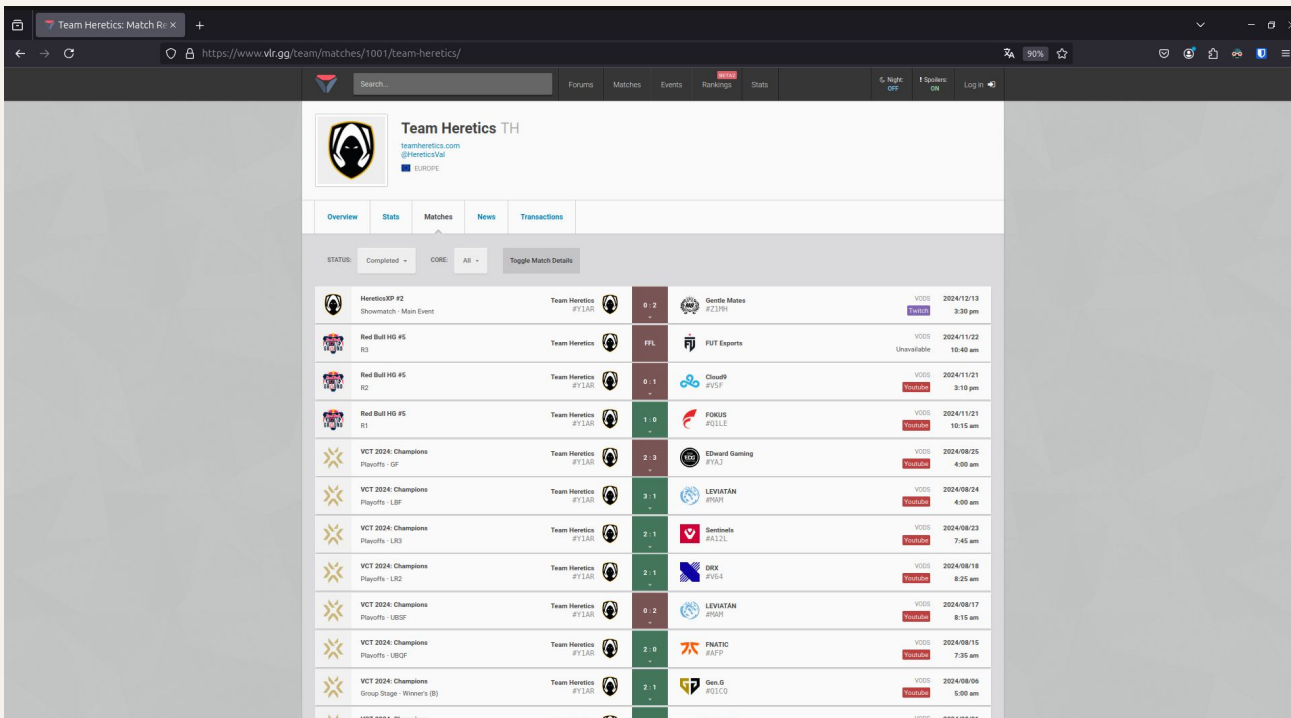
Playoffs - 2nd
\$400,000

Champions Tour 2024: EMEA Stage 2

2024

Playoffs - 3rd
\$40,000












































































































Página de resultados de um time



The screenshot shows the Team Heretics website on a web browser. The browser's address bar displays the URL <https://www.vlr.gg/team/matches/1001/team-heretics/>. The website header includes navigation links for Search, Forums, Matches, Events, Rankings, and Stats. The main content area features the Team Heretics logo and social media handles. Below this, there are tabs for Overview, Stats, Matches, News, and Transactions. The 'Matches' tab is selected, showing a list of matches with filters for Status (Completed, CORE, All) and a 'Toggle Match Details' button. The matches are listed in a table with columns for match details, team names, scores, and dates.

Match Details	Team Heretics #1	Score	Team Heretics #2	Date
HereticsXP #2 Showmatch - Main Event	Team Heretics #1	0 : 2	Gentle Mates #2306	2024/12/13 3:30 pm
Red Bull HGS #5 R3	Team Heretics #1	FFL	FUT Exports	2024/11/22 10:40 am
Red Bull HGS #5 R2	Team Heretics #1	0 : 1	Cloud9 #1037	2024/11/21 3:10 pm
Red Bull HGS #5 R1	Team Heretics #1	1 : 0	FORUS #1111	2024/11/21 10:15 am
VCT 2024: Champions Playoffs - GF	Team Heretics #1	2 : 3	Edward Gaming #163	2024/09/26 4:00 am
VCT 2024: Champions Playoffs - LSF	Team Heretics #1	3 : 1	LEVITAN #1001	2024/08/24 4:00 am
VCT 2024: Champions Playoffs - LR3	Team Heretics #1	2 : 1	Sentinel #1121	2024/08/23 7:45 am
VCT 2024: Champions Playoffs - LR2	Team Heretics #1	2 : 1	DKR #1014	2024/08/18 8:25 am
VCT 2024: Champions Playoffs - USGF	Team Heretics #1	0 : 2	LEVITAN #1001	2024/08/17 8:15 am
VCT 2024: Champions Playoffs - USOF	Team Heretics #1	2 : 0	FNATIC #1017	2024/08/16 7:35 am
VCT 2024: Champions Group Stage Winner's (B)	Team Heretics #1	2 : 1	Gen.G #1010	2024/08/06 5:00 am
VCT 2024: Champions	Team Heretics #1	2 : 1	Gen.G #1010	2024/08/01

Página de estatísticas de um time

MAP (f)	EXPAND	WIN%	W	L	ATK TST	DEF TST	ATK RANK	RW	RL	DEF RANK	RW	RL	AGENT COMPOSITIONS
Bld (54)		68%	38	18	25	31	60%	371	250	50%	301	298	(9)      (1)      (1)      (3)      (5)      (1)      (3)      (1)      (3)      (1)      (11)      (4)      (1)      (1)      (4)      (1)      (3)     
Heaven (43)		75%	47	16	32	30	58%	380	271	59%	357	247	(1)      (2)      (1)      (2)     

3

Estruturação do Projeto

Infraestrutura - VLRGG Scraper

- Utilizou-se a linguagem *Python* (v3.12) para o desenvolvimento da ferramenta;
- Bibliotecas utilizadas:
 - *threading*: nativa do *Python*. Necessária para implementação do download paralelo a extração de dados, otimizando o desempenho do sistema.
 - *cloudscraper*: necessária para lidar com problema do CloudFare, que barrava o bot, mesmo se a página não estivesse listada no robots.txt.

Abordagem e métodos usados - VLRGG Scraper

- Os dados foram extraídos do site *vlr.gg* e salvos em um JSON formatado.
- Os dados podem ser de interesse de algumas entidades, como Organizações de eSports, para análise de adversários, scouting de jogadores e técnicos, ou para melhorar sua estratégia de jogo com base em estatísticas e desempenho, treinadores e jogadores, para planejar estratégias com base nos mapas e performances de outros times ou analisar os próprios dados para melhorias, apostadores e plataformas de apostas, para o cálculo de probabilidade de vitória, desenvolvedores de ferramentas e aplicativos, para integrar os dados em apps voltados para eSports, como assistentes de treino, ferramentas de scouting, ou aplicativos de previsão, etc.

Abordagem e métodos usados - VLRGG Scraper

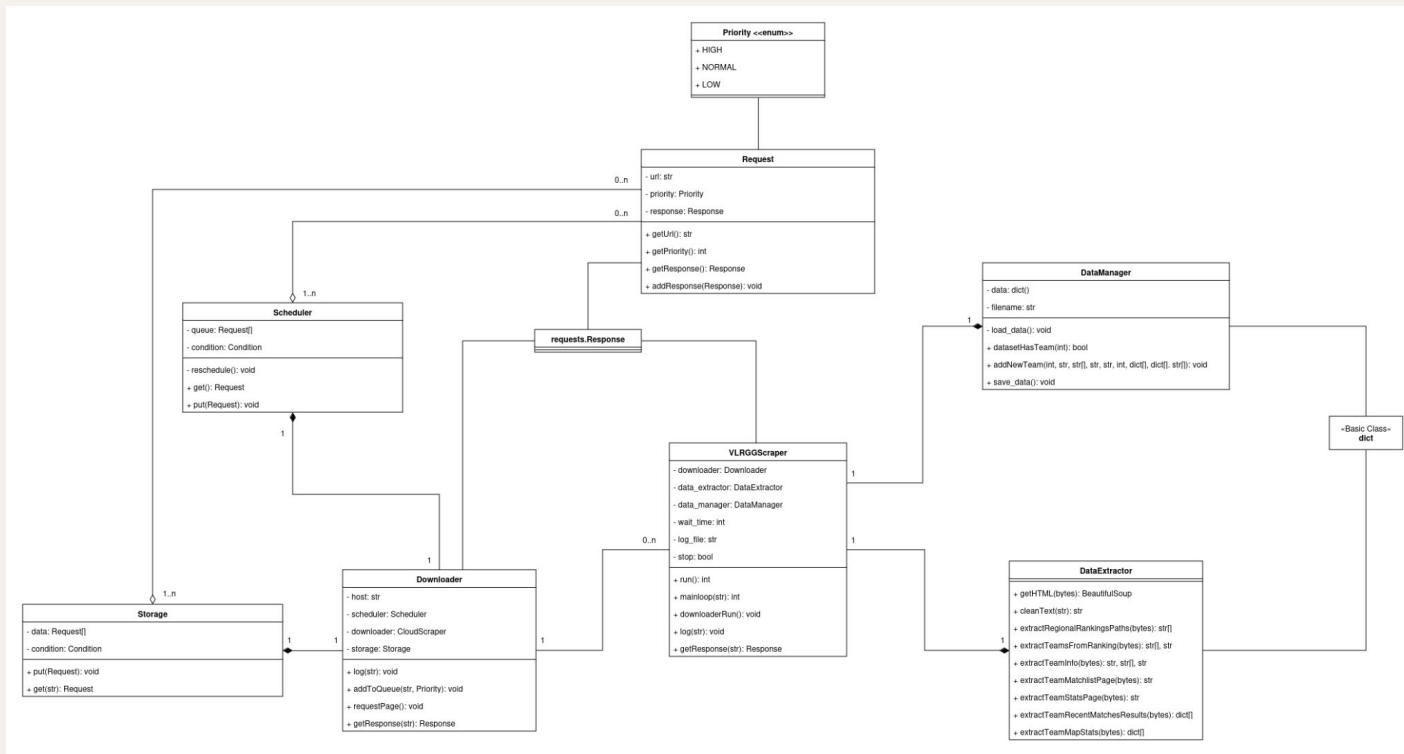
- O idioma considerado na extração dos dados foi o Inglês Norte Americano.
- O *dataset* gerado é composto por uma lista de times, onde cada time terá os seguintes atributos:
 - Id
 - Nome
 - Jogadores
 - Técnico (Head Coach)
 - Região
 - Posição no ranking regional
 - Histórico de partidas recentes (até 6 meses atrás)
 - Estatísticas de mapa, onde cada mapa terá uma porcentagem de vitórias, porcentagem de rounds vencidos do lado ataque e porcentagem de rounds vencidos do lado defesa.
 - URLs utilizadas na extração de informações do time

Abordagem e métodos usados - VLRGG Scraper

```
1 VLRGG_Scraping_Dataset.json x
2 VLRGG_Scraping_Dataset.json > ...
3 {
4   "teams": [
5     {
6       "Id": "1001",
7       "Name": "Team Heretics",
8       "Players": [
9         "Ricardas 'Boo' Lukasevicius",
10        "Dominykas 'MiniBoo' Lukasevicius",
11        "Wart 'Wart' Alkan",
12        "Enes 'RieNs' Ecirli",
13        "Benjamin 'benjifishy' David 'benjifishy' Fish"
14      ],
15      "Coach": "Neli 'neilzinho' Finlay",
16      "Region": "Europe",
17      "Rank": 1,
18      "Recent Results": [
19        {
20          "date": "2024/12/13",
21          "series": "b03",
22          "opponent": "Gentle Mates",
23          "result": "loss",
24          "maps": {
25            "Haven": "10-13",
26            "Abyss": "11-13"
27          }
28        },
29        {
30          "date": "2024/11/22",
31          "series": "",
32          "opponent": "FUT Esports",
33          "result": "loss",
34          "maps": {}
35        },
36        {
37          "date": "2024/11/21",
38          "series": "b01",
39          "opponent": "Cloud9",
40          "result": "loss",
41          "maps": {
42            "Abyss": "11-13"
43          }
44        },
45        {
46          "date": "2024/11/21",
47          "series": "b01",
48          "opponent": "FOWUS",
49          "result": "win",
50          "maps": {}
51        }
52      ]
53    }
54  ]
55 }
```

```
1 VLRGG_Scraping_Dataset.json x
2 VLRGG_Scraping_Dataset.json > ...
3 {
4   "teams": [
5     {
6       "Id": "1001",
7       "Name": "Team Heretics",
8       "Players": [
9         "Ricardas 'Boo' Lukasevicius",
10        "Dominykas 'MiniBoo' Lukasevicius",
11        "Wart 'Wart' Alkan",
12        "Enes 'RieNs' Ecirli",
13        "Benjamin 'benjifishy' David 'benjifishy' Fish"
14      ],
15      "Coach": "Neli 'neilzinho' Finlay",
16      "Region": "Europe",
17      "Rank": 1,
18      "Recent Results": [
19        {
20          "date": "2024/12/13",
21          "series": "b03",
22          "opponent": "Gentle Mates",
23          "result": "loss",
24          "maps": {
25            "Haven": "10-13",
26            "Abyss": "11-13"
27          }
28        },
29        {
30          "date": "2024/11/22",
31          "series": "",
32          "opponent": "FUT Esports",
33          "result": "loss",
34          "maps": {}
35        },
36        {
37          "date": "2024/11/21",
38          "series": "b01",
39          "opponent": "Cloud9",
40          "result": "loss",
41          "maps": {
42            "Abyss": "11-13"
43          }
44        },
45        {
46          "date": "2024/11/21",
47          "series": "b01",
48          "opponent": "FOWUS",
49          "result": "win",
50          "maps": {}
51        }
52      ]
53    }
54  ]
55 }
```


Diagrama de classes - VLRGG Scraper



Implementação - VLRGG Scraper

- Método *run*, classe *VLRGGScraper*:
 - Inicializa as threads para download das páginas;
 - Adiciona a página com os rankings regionais a fila do *downloader*;
 - Espera pela resposta no *getResponse*, e, se for uma resposta válida, passa para o extrator de dados, que retorna as URLs dos rankings regionais.
 - Adiciona essas URLs a fila do *downloader* e, para cada URL (ranking), efetua a chamada do *mainloop*.

```
11 class VLRGGScraper:
12     def run(self):
13         ...
14         Inicializa o WebScraper, configurando uma Thread dedicada para o crawler e
15         efetuando o loop de execução para cada ranking.
16         ...
17         # Configurando thread crawler
18         for i in range(5):
19             crawler_thread = Thread(target=self.downloaderRun, daemon=True)
20             crawler_thread.start()
21
22         # 1. Extrair uma lista de rankings por região
23         self.downloader.addToQueue("/rankings", 1)
24
25         # Obtém a resposta da requisição
26         response = self.getResponse("/rankings")
27         if not response:
28             self.log("Invalid response for URL: '/rankings' (ABORTING)")
29             return -1
30
31         # O método retorna uma lista com a url de todos os rankings por região
32         self.log("Extracting Regional Rankings URLs...")
33         try:
34             rankings = self.data_extractor.extractRegionalRankingsPaths(response)
35         except Exception as e:
36             self.log("Regional Rankings URLs extraction FAILED with error [e] (ABORTING)")
37             return -1
38
39         # rankings = ["/rankings/brazil"]
40
41         # 2. Adicionando o link dos rankings na fila de requisições
42         self.log("Adding rankings URLs to queue...")
43         for ranking in rankings:
44             self.downloader.addToQueue(ranking, 1)
45
46         # 3. Para cada ranking, executa o loop
47         for ranking in rankings:
48             self.log("-" * 70)
49             self.log(f"Starting ranking '{ranking}' data extraction...")
50
51             code = self.mainloop(ranking)
52             if code == -1:
53                 self.log(f"Data extraction for ranking '{ranking}' finished with errors!")
54                 continue
55
56             self.log(f"Data extraction for ranking '{ranking}' successfully finished!")
57             self.log("-" * 70)
58
59         self.data_manager.save_data()
60         self.stop = True
61         return 0
```

Implementação - VLRGG Scraper

```
scraper.py X
src > scraper.py > VLRGGScraper > mainloop
11 class VLRGGScraper:
108 def mainloop(self, ranking):
109     """
110     Método principal do sistema.
111
112     Para cada ranking:
113         - Extrai os times e a região do ranking
114         - Para os 100 primeiros times do ranking
115         - Extrai as informações citadas na descrição da classe
116     """
117     return_code = 0
118
119     # 1. Espera pela resposta da requisição ao ranking
120     response = self.getResponse(ranking)
121     if not response:
122         self.log(f"Invalid response for URL '{ranking}'! (SKIPPING RANKING)")
123         return -1
124
125     # 2. Extrai as URLs dos times
126     self.log(f"Extracting Teams URLs from ranking '{ranking}'...")
127     try:
128         urls, region = self.data_extractor.extractTeamsFromRanking(response)
129     except Exception as e:
130         self.log(f"Teams URLs extraction from ranking '{ranking}' FAILED with error (e)! (SKIPPING RANKING)")
131         return -1
132
133     # 3. Adiciona as URLs na fila de requisições
134     for url in urls:
135         self.downloader.addToQueue(url, 2)
136
```

- Método *mainloop*, classe *VLRGGScraper* (1):
 - Recebe a URL de um ranking como parâmetro;
 - Aguarda pela resposta da requisição a esse ranking;
 - Se a resposta for válida, repassa ao extrator;
 - O extrator retorna as URLs de todos os times no ranking e a região desse ranking;
 - Adiciona essas URLs na fila do *downloader*;

Implementação - VLRGG Scraper

- Método *mainloop*, classe *VLRGGScraper* (2):
 - Para cada URL (time):
 - Espera até receber a resposta da requisição com a URL da página do time;
 - Se a resposta for válida, primeiro verifica se o time já existe no *dataset*. Caso não exista, repassa a resposta ao extrator, que extrai e retorna o nome, os jogadores e o técnico (se houver) do time;
 - Depois, envia a mesma resposta novamente ao extrator, que extrai e retorna as páginas de partidas e estatísticas do time (nessa ordem). Essas páginas são adicionadas a fila;

```
scraper.py X
src > scraper.py VLRGGScraper > @ mainloop
11 class VLRGGScraper:
12     def mainloop(self, ranking):
13         # 4. Para cada URL (time) no ranking
14         rank = 0
15
16         for url in urls:
17             team_urls = [self.host+ranking, self.host+url]
18             rank += 1
19
20             self.log(f"Starting team '{url}' data extraction...")
21
22             # 4.1 Espera pela resposta da requisição da página do time
23             response = self.getResponse(url)
24             if not response:
25                 self.log(f"Invalid response for URL '{url}'! (SKIPPING TEAM)")
26                 return_code = -1
27                 continue
28
29             # 4.2 Extraí o ID do time da URL
30             team_id = re.search("(id=)", url).group(0)
31             if self.data_manager.datasetHasTeam(team_id):
32                 self.log(f"Team '{team_id}' already on the dataset! (SKIPPING TEAM)")
33                 continue
34
35             # 4.3 Extraí nome, jogadores e head coach do time
36             self.log(f"Extracting team '{url}' info...")
37             try:
38                 name, players, coach = self.data_extractor.extractTeamInfo(response)
39             except Exception as e:
40                 self.log(f"Team '{url}' info extraction FAILED with error {e}! (SKIPPING TEAM)")
41                 return_code = -1
42                 continue
43
44             # 4.4 Extraí a URL para a página com a lista de partidas do time e adiciona na fila de requisições
45             self.log(f"Extracting team '{url}' matchlist page URL...")
46             try:
47                 matchlist_page = self.data_extractor.extractTeamMatchlistPage(response)
48             except Exception as e:
49                 self.log(f"Team '{url}' matchlist page URL extraction FAILED with error {e}! (SKIPPING TEAM)")
50                 return_code = -1
51                 continue
52             self.downloader.addToQueue(matchlist_page, 3)
53
54             # 4.5 Extraí a URL para a página de estatísticas do time e adiciona na fila de requisições
55             self.log(f"Extracting team '{url}' statistics page URL...")
56             try:
57                 stats_page = self.data_extractor.extractTeamStatsPage(response)
58             except Exception as e:
59                 self.log(f"Team '{url}' statistics page URL extraction FAILED with error {e}! (SKIPPING TEAM)")
60                 return_code = -1
61                 continue
62             self.downloader.addToQueue(stats_page, 3)
```

Implementação - VLRGG Scraper

```
scraper.py M X
scr > scraper.py > % VLRGGScraper > @ mainloop
11 class VLRGGScraper:
108 def mainloop(self, ranking):
189     # 4.6 Espera pela resposta da requisição da página de lista de partidas do time
190     response = self.getResponse(matchlist_page)
191     if not response:
192         self.log(f"Invalid response for URL '{matchlist_page}'! (SKIPPING TEAM)")
193         return_code = -1
194         continue
195     team_urls.append(self.host+matchlist_page)
196
197     # 4.7 Extraí os resultados recentes do time
198     self.log(f"Extracting team '{url}' recent matches results...")
199     try:
200         recent_results = self.data_extractor.extractTeamRecentMatchesResult(response)
201     except Exception as e:
202         self.log(f"Team '{url}' recent matches results extracion FAILED with error (e)! (SKIPPING TEAM)")
203         return_code = -1
204         continue
205
206     # 4.8 Espera pela resposta da requisição da página de estatísticas do time
207     response = self.getResponse(stats_page)
208     if not response:
209         self.log(f"Invalid response for URL '{stats_page}'! (SKIPPING TEAM)")
210         return_code = -1
211         continue
212     team_urls.append(self.host+stats_page)
213
214     # 4.9 Extraí as estatísticas de mapas do time
215     self.log(f"Extracting team '{url}' maps statistics...")
216     try:
217         maps_stats = self.data_extractor.extractTeamMapsStats(response)
218     except Exception as e:
219         self.log(f"Team '{url}' maps statistics extraction FAILED with error (e)! (SKIPPING TEAM)")
220         return_code = -1
221         continue
222
223     # 4.10 Salva o time na na lista de times
224     self.data_manager.addNewTeam(team_id, name, players, region, coach, rank, recent_results, maps_stats, team_urls)
225     self.log(f"Team '{name}' added to dataset!")
226
227     self.log(f"~*~*~")
228
229     return return_code
```

- Método *mainloop*, classe *VLRGGScraper* (3):
 - Para cada URL (time) [continuação do loop]:
 - Espera pela resposta da requisição realizada a página de partidas do time;
 - Se a resposta for válida, repassa ao extrator, que extrai e retorna as informações sobre partidas recentes (últimos 6 meses) do time;
 - Faz o mesmo processo para a página de estatísticas, mas, dessa vez, o extrator retorna estatísticas históricas do desempenho do time por mapas;
 - Por fim, adiciona o time ao *dataset*;

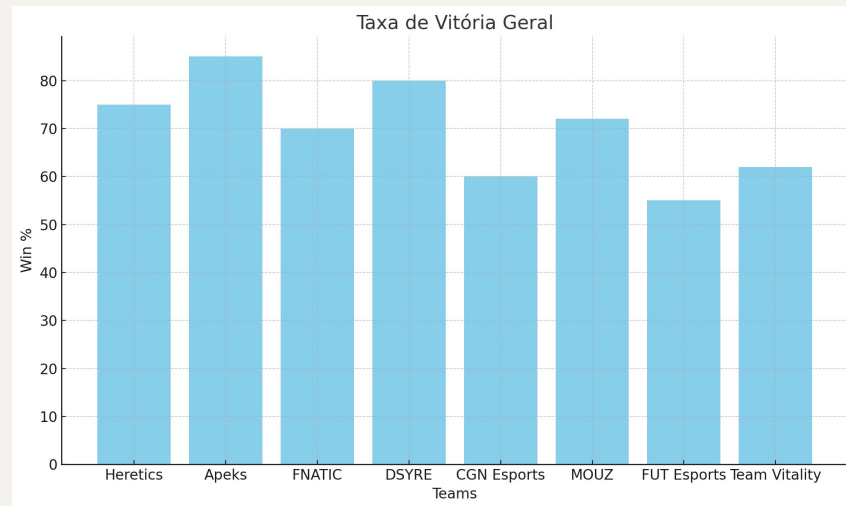
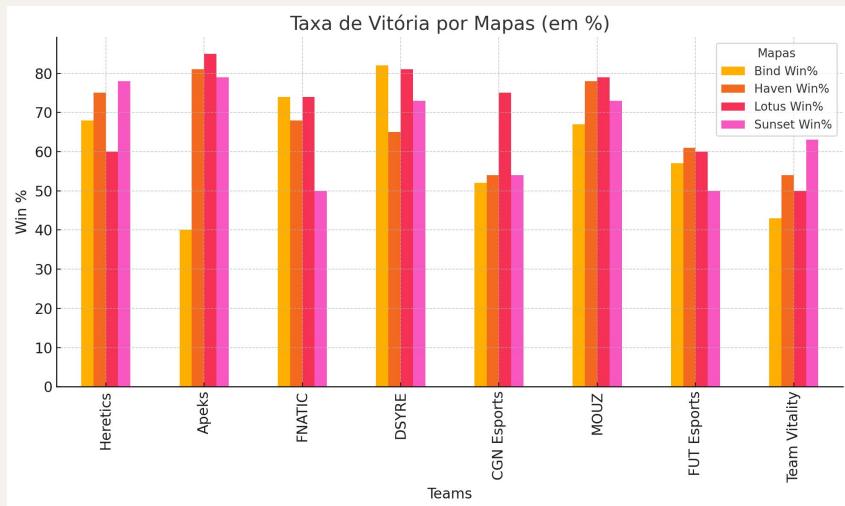
4

Análise dos resultados finais

Visão Geral

- Os dados extraídos oferecem uma análise abrangente do desempenho dos times do cenário competitivo de *Valorant*, incluindo vitórias e derrotas recentes, estatísticas detalhadas por mapa e desempenhos regionais. As interpretações podem ser apresentadas em gráficos como taxas de vitória por mapa, tendências de resultados ao longo do tempo e comparações regionais, destacando mapas fortes e fracos, além de diferenças de desempenho entre ataque e defesa. Essas análises permitem identificar padrões de desempenho e posicionamento competitivo, auxiliando na avaliação estratégica dos times.

Exemplos de uso



Exemplos de uso

- Além dos gráficos, é possível utilizar os dados no cálculo de probabilidade de vitórias, como comentado anteriormente. As imagens ao lado apresentam uma aplicação simples, desenvolvida, também, em *Python*, que utiliza os dados para essa simulação. Não é um modelo muito robusto, mas é suficiente para demonstrar a utilização dos dados extraídos.

```
=====
                        Valorant Matchup Simulator
=====
Input team A name: Team Heretics
-----
Input team B name: Leviatán
-----
Number of maps played (series): 2
Invalid number of maps! Must be 1, 3 or 5.
-----
Number of maps played (series): 3
-----
Map 1: Bind
Map 2: Ascent
Map 3: Haven
-----
Probability of team A winning: 52.8%
Probability of team B winning: 47.2%
=====
```

```
=====
                        Valorant Matchup Simulator
=====
Input team A name: 2Game Esports
-----
Input team B name: Team Liquid
-----
Number of maps played (series): 1
-----
Map 1: Sunset
-----
Probability of team A winning: 82.3%
Probability of team B winning: 17.7%
=====
```

5

Considerações Finais

Dificuldades e Falhas

- Ideia inicial era extrair os dados de vetos dos times do portal *HLTV.org* (parecido com o *vlr.gg*, porém para o jogo *Counter Strike*);
 - Falha inicial no scraper por conta de um bloqueio do CloudFare → necessitou o uso da lib *cloudscraper*;
 - Ranking global com poucos times (250-300);
 - Passou por atualização recente que bloqueou o acesso às páginas de resultados de partidas por *bots*;
- Alteração da fonte de dados para o VLRGG;
 - Rankings regionais com quase todos os times (mais de 1000 no total);
 - Todas as páginas acessíveis por *bots*;
 - Porém informações de vetos nem sempre são disponibilizadas nas páginas de resultados de uma partida;
 - Por isso a opção de extrair dados estatísticos de vitórias e derrotas, e resultados por mapa, dados que, em 99% dos casos, estão disponíveis.

Conclusão

- Atualmente, a extração ocorre para 1581 times. Todo o processo de extração leva de 3 a 4 horas. Maneiras de melhorar esse desempenho seriam:
 - Uso de processos concorrentes com proxies rotativos;
 - Uso da biblioteca Scrapy: possui paralelismo nativo, mas não diferencia os IPs de acesso.
- O trabalho se mostrou desafiador e consolidou bem o aprendizado de conceitos relacionados à extração e tratamento de dados.

