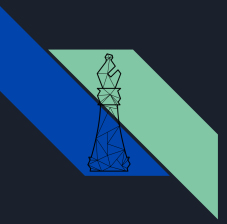
BispOSet: Chess Match and Titled Player Statistic Organized Dataset

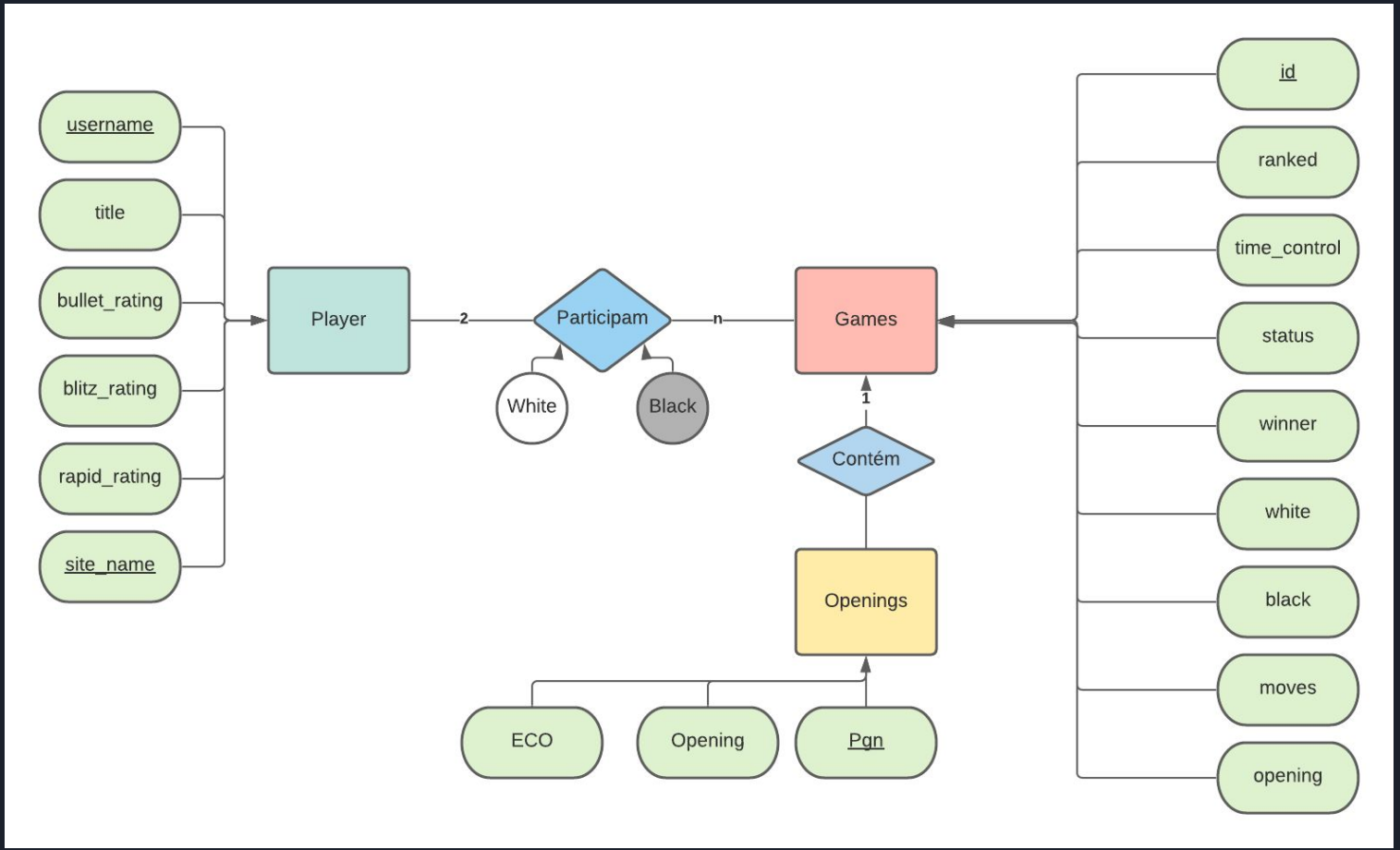
- Breno Nunes Tavares RA: 232238
- Eduardo Carvalheira Teixeira de Aguiar RA:233686
- Enrico Piovesana Fernandes RA: 233895



Contexto/Motivação

- O dataset foi criado com a motivação de criar um banco de dados expansivo e unificado que compreende o jogo de xadrez jogado por jogadores com títulos formais.
- O dataset busca reunir informações sobre vários aspectos e facetas do xadrez, como ranks, número de jogos, estatística e informações sobre as partidas jogadas.

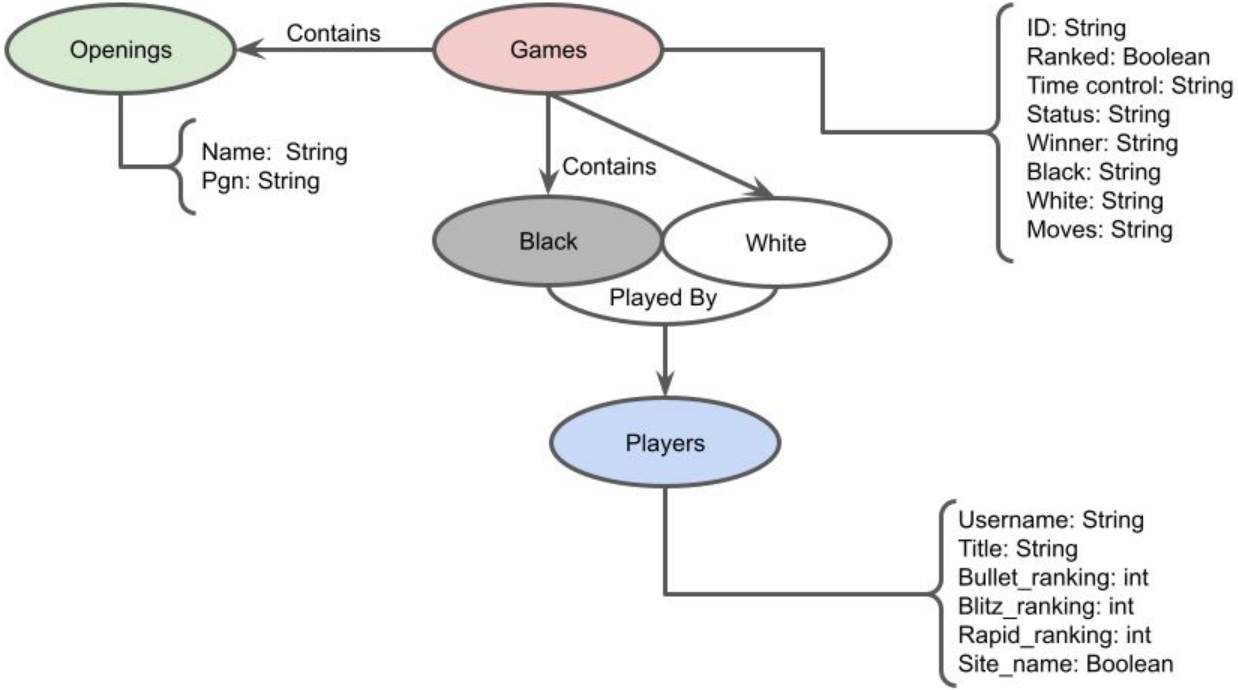
Modelo Conceitual



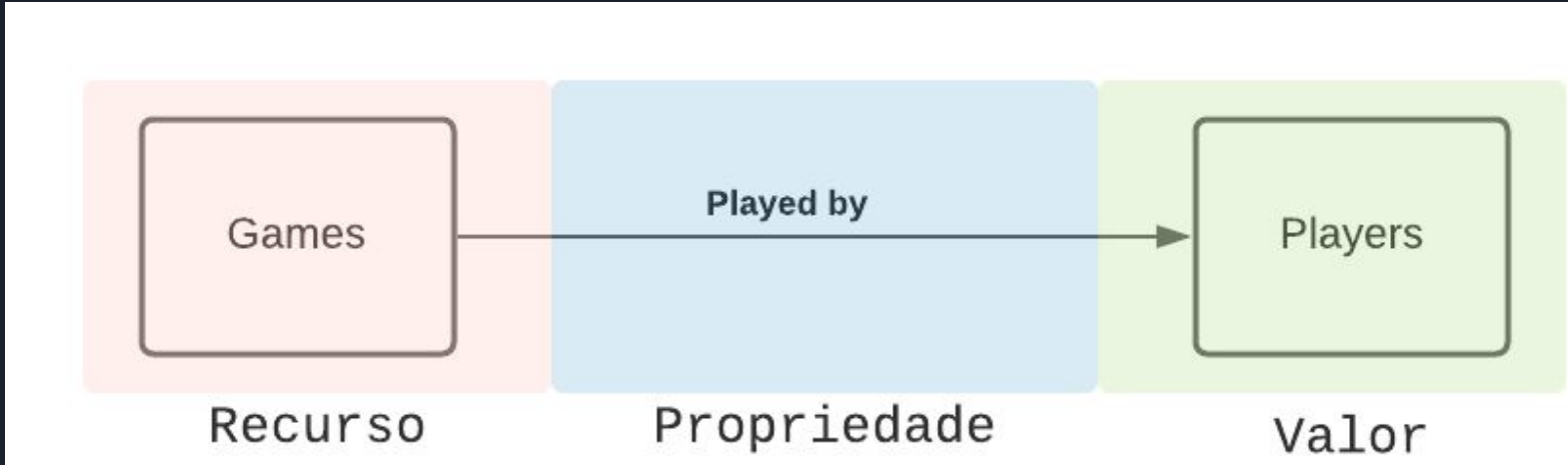
Modelos Lógicos

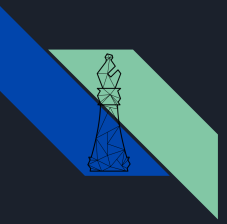
- Os dados seriam organizados de uma forma fácil e compreensiva, sendo divididos entre diversos arquivos .csv, com associações sendo feitas a partir dos IDs únicos dos jogadores.
- O modelo lógico utilizado seriam tabelas.

Modelo Relacional



Grafo de Conhecimento





Exemplos de perguntas que podem ser respondidas pelo dataset

- Qual a abertura mais comum entre os jogadores que possuem um título específico?
- Qual a casa onde o rei toma mais xeque-mates?
- Qual a casa onde mais peões promovem?
- Qual a peça que é comida mais rapidamente em geral?
- Qual a porcentagem de partidas com roque curto/longo/partidas sem roque?
- Qual o número médio de partidas entre jogadores de um título específico?

Fontes de dados e ferramentas utilizadas

- https://lichess.org/api - API
- https://lichess.org/blog/X0I7phAAACQAMCfH/titled-arena-announcements - extração de dados de Página Web

```
player_df = pd.DataFrame(columns=['username','title','bullet_rating','blitz_rating','rapid_rating'])
```
[5] Python

```
for curr_tag in titled_tournaments['tag']:
    tour_results = client.tournaments.stream_results(curr_tag)
    tour_json = list(tour_results)
    for i in tour_json:
        try:
            temp_df = pd.DataFrame({'username':[i['username']],'title':[i['title']],'bullet_rating':[0],'blitz_rating':[0],'r
            player_df = pd.concat([player_df,temp_df])
        except:
            test_ratings = 0
```
[6] Python

Lichess player fetcher

```
player_df
```

	username	title	bullet_rating	blitz_rating	rapid_rating
0	alireza2003	GM	3126	2823	1756
1	penguingim1	GM	2856	2828	2555
2	Night-King96	GM	2984	2857	0
3	Leon_Livaic	IM	0	0	0
4	Vladimirovich9000	GM	2960	2900	0
...	...	...	...	...	...
4789	B_nn_t	FM	2280	2422	2130
4790	royalblue04	FM	1977	2307	2426
4791	hAnN18Al	FM	2137	2081	0
4792	Bwert	CM	2122	2188	1675
4793	Pavelkost87	FM	1722	2495	2172

4794 rows × 5 columns

```
start = 27
end = 30
for j in players[start:end]['username']:
    currgames = list(client.games.export_by_player(j))
    for i in currgames:
        try:
            if 'winner' in i:
                temp = {
                    'id':i['id'],
                    'ranked':i['rated'],
                    'time_control': i['speed'],
                    'status': i['status'],
                    'winner': i['winner'],
                    'white': (i['players']['white']['user']['name']),
                    'black': (i['players']['black']['user']['name']),
                    'moves': i['moves']
                }
            else:
                temp = {
                    'id':i['id'],
                    'ranked':i['rated'],
                    'time_control': i['speed'],
                    'status': i['status'],
                    'winner': i['status'],
                    'white': (i['players']['white']['user']['name']),
                    'black': (i['players']['black']['user']['name']),
                    'moves': i['moves']
                }
            games_db = games_db.append(temp,ignore_index=True)
        except:
```

```
,id,ranked,time_control,status,winner,white,black,moves,opening
0,HpP16h6T,True,bullet,resign,white,DrGrekenstein,hitter1999,1. Nf3 Nf6 2. g3 g6 3. Bg2 Bg7 4. O-O O-O 5. d4
1,3GmaSy1u,True,bullet,mate,white,DrGrekenstein,C9C9C9C9C9,1. e4 e6 2. d4 d5 3. e5 Nd7 4. Nf3 c5 5. c3 Ne7 6
2,TWbE2AMW,True,bullet,mate,black,toivok3,DrGrekenstein,1. d4 d5 2. c4 dxc4 3. Nf3 Nf6 4. e3 e6 5. Bxc4 a6 6
3,3pEj9qmK,True,bullet,draw,draw,DrGrekenstein,alireza2003,1. Nf3 d5 2. g3 Nf6 3. Bg2 e6 4. O-O Be7 5. b3 O-
4,URJyg0hU,True,bullet,mate,white,drop_stone,DrGrekenstein,1. e4 c6 2. d4 d5 3. Nc3 dxe4 4. Nxe4 Nf6 5. Nxf6
5,ec8YRTyb,True,bullet,resign,white,DrGrekenstein,junglebook1,1. e4 e5 2. d4 exd4 3. Qxd4 Nc6 4. Qe3 Nf6 5.
6,578lajIo,True,bullet,mate,black,Zhigalko_Sergei,DrGrekenstein,1. e4 e5 2. Nf3 Nc6 3. Bc4 Bc5 4. b4 Bxb4 5.
7,PNub5K23,True,bullet,resign,white,DrGrekenstein,C9C9C9C9C9,1. e4 e6 2. d4 d5 3. Nd2 dxe4 4. Nxe4 Nd7 5. Nf
8,vYMmXuaX,True,bullet,resign,black,junglebook1,DrGrekenstein,1. d4 Nf6 2. c4 c5 3. d5 b5 4. cxb5 a6 5. f3 g
9,0m4epfHk,True,bullet,resign,black,mutdpro,DrGrekenstein,1. d4 Nf6 2. Nf3 c5 3. c3 e6 4. Bf4 b6 5. e3 cxd4
10,Yx3URrr3,True,bullet,resign,white,DrGrekenstein,rasulovvugar,1. e3 Nf6 2. d4 g6 3. c4 Bg7 4. Nc3 O-O 5. N
11,htrkwexF,True,bullet,draw,draw,RebeccaHarris,DrGrekenstein,1. e4 c6 2. d3 d5 3. Nd2 e5 4. Ngf3 Bd6 5. g3
12,YoguzGDY,True,bullet,draw,draw,DrGrekenstein,toivok3,1. e4 e6 2. d4 d5 3. Nd2 c5 4. Ngf3 cxd4 5. exd5 Qxd
13,Wwg9YugR,True,bullet,outoftime,white,DrGrekenstein,junglebook1,1. e4 e5 2. d4 exd4 3. Qxd4 Nc6 4. Qe3 Nf6
```

Lichess game fetcher

Fontes de dados e ferramentas utilizadas

- https://www.chess.com/news/view/published-data-api - API
- https://github.com/sarartur/chess.com - Python Wrapper for chess.com API

```python
# pega o nome dos players e coloca os titulos em duas listas
player_names = []
player_titles = []

titles = ['GM', 'NM', 'FM', 'NM', 'CM', 'WGM']

for i in titles:
    response = chessdotcom.get_titled_players(i)
    player_names = player_names + response.json['players']
    for j in range(len(response.json['players'])):
        player_titles.append(i)
```

players

	username	title	bullet_rating	blitz_rating	rapid_rating
0	123lt	GM	2725	2704	2482
1	124chess	GM	2512	2713	2244
2	1977ivan	GM	2650	2707	2342
3	1stsecond	GM	2851	2655	2496
4	4thd-alpeacefulmoon	GM	0	0	0
...	...	...	...	...	...
8556	yaris9	WGM	2440	2451	2330
8557	yennefer1	WGM	2304	2436	2267
8558	yileai	WGM	0	2314	2167
8559	zefirka	WGM	2443	2463	1980
8560	zombik1	WGM	0	2372	0

8561 rows × 5 columns

Chess.com player fetcher

```python
games_db = pd.read_csv('empty_games.csv')

for k in players_to_db['username']:
    month, year = getDates(k)
    print(k)
    for i in range(len(month)):
        response = chessdotcom.get_player_games_by_month(k,month=month[i],year=year[i]).json
        for j in (response['games']):
            try:
                temp = {
                    'id':j['uuid'],
                    'ranked':j['rated'],
                    'time_control': j['time_class'],
                    'status': checkstatus(j['white']['result'],j['black']['result']),
                    'winner': checkwinner(j['white'],j['black']),
                    'white': (j['white']['username']),
                    'black': (j['black']['username']),
                    'moves': trataPGN(j['pgn'])
                }
                #print((j['white']['result'],j['black']['result']))
                games_db = games_db.append(temp,ignore_index=True)
            except:
                temp = 0
```

[21]

Chess.com game fetcher

```
games_db
```

Python

	id	ranked	time_control	status	winner	white	black	moves
0	e6cd2f50-bb4f-11dd-8000-000000010001	True	blitz	checkmated	Atalik	Atalik	ubahn	1. d4 1... e6 2. c4 2... d6 3. e4 3... c6...
1	a7564b64-bdba-11dd-8000-000000010001	True	bullet	resigned	Atalik	martinZH	Atalik	1. d4 1... Nf6 2. c4 2... e6 3. Nc3 3... ...
2	f603e53c-bdba-11dd-8000-000000010001	True	blitz	timeout	Atalik	Atalik	rock_baby_rock	1. d4 1... Nf6 2. c4 2... e6 3. Nc3 3... ...

Primeiras Consultas - Abertura mais jogada por determinado jogador

```sql
DROP TABLE IF EXISTS MaxOpenings;

CREATE VIEW MaxOpenings AS
SELECT G.Openings, COUNT(*) Aberturas
    FROM Games G, Players P
    WHERE P.Username = 'kirillgenius' AND (P.Username = G.White OR P.Username = G.Black)
    GROUP BY G.Openings
    ORDER BY COUNT(*) DESC;

SELECT * FROM MaxOpenings
```

index	OPENINGS	ABERTURAS
0	Horwitz Defense	1944
1	Sicilian Defense: French Variation	1762
2	Caro-Kann Defense	1692
3	Sicilian Defense: Modern Variations	1654
4	Sicilian Defense: Old Sicilian	1506
5	French Defense: Queen's Knight	1482
6	Pirc Defense	1182
7	French Defense: Tarrasch Variation	986
8	Sicilian Defense: Closed	900
9	Van't Kruijs Opening	882
10	English Opening: Agincourt Defense	764
11	Sicilian Defense: Four Knights Variation	734
12	Vienna Game: Stanley Variation, Three Knights Variation	610
13	Vienna Game: Max Lange Defense	590
14	Zukertort Opening: Queen's Gambit Invitation	590
15	Vienna Game: Stanley Variation	572
16	French Defense: Knight Variation	558
17	Queen's Gambit Declined	524
18	Benoni Defense: Czech Benoni Defense	490

Primeiras Consultas - Abertura mais jogada entre todos os jogadores

```sql
DROP TABLE IF EXISTS MaxOpenings;

CREATE VIEW MaxOpenings AS
SELECT G.Openings, COUNT(*) Aberturas
    FROM Games G
    GROUP BY G.Openings
    ORDER BY COUNT(*) DESC;

SELECT * FROM MaxOpenings
```

index	OPENINGS	ABERTURAS
0	Queen's Pawn Game	26674
1	Modern Defense	24245
2	Benoni Defense: Old Benoni	22307
3	Pirc Defense	18965
4	Caro-Kann Defense	17729
5	Zukertort Opening	17262
6	Mieses Opening	15551
7	Van't Kruijs Opening	14755
8	Indian Defense	14056
9	Horwitz Defense	12832
10	Sicilian Defense: Closed	12450
11	Van Geet Opening	11993
12	Scandinavian Defense	11582
13	Hungarian Opening	10656
14	Indian Defense: Knights Variation	9948
15	Zukertort Opening: Sicilian Invitation	9348
16	Zukertort Opening: Pirc Invitation	9213
17	French Defense: Queen's Knight	9072
18	Nimzo-Larsen Attack: Classical Variation	8322

Primeiras Consultas - Win Rate de um jogador específico

```sql
SELECT P.Username, WW.GanhouDeBrancas, WB.GanhouDePretas, TB.JogosDePretas, TW.JogosDeBrancas,
((CAST(WW.GanhouDeBrancas+WB.GanhouDePretas AS float))/(TB.JogosDePretas+TW.JogosDeBrancas)) AS Winrate
FROM WinnerW WW, WinnerB WB, TotalBlack TB, TotalWhite TW, Players P
WHERE P.Username = 'umutcrazy'
```

```sql
DROP TABLE IF EXISTS WinnerB;

CREATE VIEW WinnerB AS
SELECT P.Username, COUNT(*) GanhouDePretas
    FROM Games G, Players P
    WHERE G.Winner = 'black' AND G.Black = P.Username AND G.SiteName = 'lichess' AND P.Username = 'umutcrazy'
    GROUP BY P.Username
    ORDER BY COUNT(*) DESC;

SELECT WB.GanhouDePretas FROM WinnerB WB
```

✓ 0.8s

27451

index	Key	Value
0	USERNAME	umutcrazy
1	GANHOUDEBRANCAS	28599
2	GANHOUDEPRETAS	27451
3	JOGOSDEPRETAS	38547
4	JOGOSDEBRANCAS	38505
5	WINRATE	0.7274308259357317

Primeiras Consultas - Jogadores com mais vitórias

```
SELECT X.winner, COUNT (*) WINS
FROM (SELECT G.result, G.winner,G.white,G.black
FROM Games G
WHERE G.winner = G.white OR G.winner = G.Black) X
GROUP BY X.winner
ORDER BY WINS DESC;
```

index	WINNER	WINS
0	umutcrazy	56050
1	OhanyanEminChess	33465
2	MeneerMandje	22757
3	Niclox	21084
4	sumsar42	16904
5	toivok3	16431
6	Zhigalko_Sergei	15305
7	Arka50	13930
8	kirillgenius	13809
9	samisahi	13622
10	Bernd_Stromberg	13446
11	nochewycandy	13059
12	Blazinq	11900
13	ChessShop	11763
14	backreg	11279
15	PakMaster	10482
16	JonahWillow	10418
17	defibaugh	10360
18	shnitez	10198
19	chessforthesun	10000
20	HouseMartell	9915