

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

GILLIARD GABRIEL RODRIGUES

MONOGRAFIA EM SISTEMAS DE INFORMAÇÃO

**Técnicas de Mineração de Padrões Frequentes aplicadas no contexto de Formação
de Equipes para o Pokémon Showdown**

PESQUISA MISTA

Belo Horizonte
2023 / 2º semestre

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Bacharelado em Sistemas de Informação

**TÉCNICAS DE MINERAÇÃO DE PADRÕES
FREQUENTES APLICADAS NO CONTEXTO DE
FORMAÇÃO DE EQUIPES PARA O
POKÉMON SHOWDOWN**

por

GILLIARD GABRIEL RODRIGUES

Monografia em Sistemas de Informação

Apresentado como requisito da disciplina de Monografia em
Sistemas de Informação do Curso de Bacharelado em Sistemas de
Informação da UFMG

Prof. Dr. *Flávio Vinícius Diniz de Figueiredo*
Orientador(a)

Belo Horizonte
2023 / 2º semestre

Dedico este trabalho a todos que sempre me apoiaram ao longo dessa caminhada, em especial à minha mãe, Marilene Rodrigues e à minha amiga, Rafaela Taiane.

AGRADECIMENTOS

Inicialmente agradeço à minha mãe, Marilene Rodrigues, por seu amor e por sempre me incentivar nos estudos, determinada a mover montanhas para me prover as melhores oportunidades apesar das difíceis circunstâncias.

Agradeço à minha amiga, Rafaela Taiane, pelo seu apoio em meus objetivos pessoais e profissionais. Aos meus professores, pelos conhecimentos adquiridos. E, finalmente, aos colegas de curso pela convivência e trocas.

"A educação é o nosso passaporte para o futuro,
pois o amanhã pertence às pessoas
que se preparam hoje."
Malcolm X

RESUMO

O principal objetivo do presente trabalho é demonstrar a utilidade de técnicas de mineração de dados no contexto de Pokémon, explorando o uso de técnicas de mineração de padrões frequentes como uma ferramenta para auxiliar jogadores iniciantes na construção de equipes de pokémons confiáveis para batalhas competitivas da *tier OverUsed* – categoria popular na comunidade *Smogon* –, baseado em equipes já utilizadas por jogadores reais e assíduos, mineradas através de *web scraping*, bem como explorar a versatilidade de redes complexas para evidenciar relacionamentos entre pokémons utilizados em equipes com frequência, além da popularidade de cada monstinho no competitivo. Em paralelo a isso, essa obra busca também explorar uma base contendo informações sobre todos os pokémons da 1ª à 9ª geração, a fim de responder questões levantadas sobre a relação de tipo, força, resistência, combinações de tipos, bem como a distribuição dos pokémons ao longo dessas categorias, minerando curiosidades únicas que dificilmente seriam descobertas sem uma análise exploratória.

Palavras-chave: pokémon, *Smogon*, *Pokémon Showdown*, força, tipo, resistência, grafo, geração.

ABSTRACT

The main objective of the present work is to demonstrate the usefulness of data mining techniques in the context of Pokémon, exploring the use of frequent pattern mining techniques as a tool to assist beginner players in building teams of reliable Pokémon for competitive battles in the tier *OverUsed* – a popular category in the *Smogon* community –, based on teams already used by real and regular players, mined through web scraping, as well as exploring the versatility of complex networks to show relationships between Pokémon used in teams frequently, in addition to the popularity of each competitive little monster. In parallel to this, this work also seeks to explore a base containing information about all Pokémon from the 1st to the 9th generation, in order to answer questions raised about the type, strength, resistance, combinations of types, as well as the distribution of Pokémon along these categories, mining unique curiosities that would hardly be discovered without an exploratory analysis.

Keywords: pokémon, *Smogon*, *Pokémon Showdown*, strength, type, resistance, graph, generation.

LISTA DE FIGURAS

FIGURA 1 – FP TREE.....	9
FIGURA 2 – FP TREE PROJETADA PARA O ITEM D.....	10
FIGURA 3 – PSEUDOCÓDIGO DO ALGORITMO <i>FP-GROWTH</i>	10
FIGURA 4 - GRAFO NÃO DIRECIONADO REPRESENTANDO A RELAÇÃO DE AMIZADE ENTRE PESSOAS	11
FIGURA 5 - AMOSTRA RETIRADA DO FÓRUM TÉCNICO DA <i>SMOGON</i> EM FORMATO <i>TXT</i>	14
FIGURA 6 - 10 POKÉMONS MAIS UTILIZADOS (<i>GEN 8</i> OU)	15
FIGURA 7 - 10 POKÉMONS MAIS UTILIZADOS (<i>GEN 9</i> OU)	15
FIGURA 8 - SUGESTÕES DO ALGORITMO PARA O SEGUNDO POKÉMON	18
FIGURA 9 - SUGESTÕES DO ALGORITMO PARA O TERCEIRO POKÉMON	18
FIGURA 10 - SUGESTÕES DO ALGORITMO PARA O QUARTO POKÉMON	19
FIGURA 12 - SUGESTÕES DO ALGORITMO PARA O 6º POKÉMON.....	19
FIGURA 13 - VISUALIZAÇÃO DO GRAFO (<i>GEN 8</i> OU).....	22
FIGURA 14 - VISUALIZAÇÃO DO GRAFO (<i>GEN 9</i> OU).....	23
FIGURA 16 - NÚMERO DE POKÉMONS POR TIPO.....	24
FIGURA 17 - NÚMERO DE POKÉMONS LENDÁRIOS/MÍTICOS POR TIPO	25
FIGURA 18 - COMBINAÇÕES DE TIPOS MAIS COMUNS.....	26
FIGURA 19 - RANKING DE FORÇA POR TIPO.....	26
FIGURA 21 - POKÉMONS MAIS RESISTENTES A UMA VARIEDADE DE TIPOS.....	28
FIGURA 22 - RANKING DE FORÇA POR GERAÇÃO	29
FIGURA 23 - VISÃO DETALHADA DA FORÇA POR GERAÇÃO	29
FIGURA 24 - SISTEMA WEB COM AS SUGESTÕES APRESENTADAS AO USUÁRIO	31
FIGURA 25 - SISTEMA WEB COM A EQUIPE FINAL FORMADA	31
FIGURA 26 - SISTEMA WEB COM A SEÇÃO DE FAQ À MOSTRA.....	32

LISTA DE QUADROS

QUADRO 1. BASE DE DADOS D COM ITENS EM ORDEM DECRESCENTE POR TRANSAÇÃO	8
---	---

LISTA DE SIGLAS

FP	<i>Frequent Pattern</i>
IA	<i>Inteligência Artificial</i>
OU	<i>OverUsed</i>
HP	<i>Health Points</i>
API	<i>Application Programming Interface</i>
GCP	<i>Google Cloud Platform</i>
FAQ	<i>Frequent Asked Questions</i>

SUMÁRIO

RESUMO	VI
ABSTRACT	VII
LISTA DE FIGURAS	VIII
LISTA DE QUADROS	IX
LISTA DE SIGLAS	X
1 INTRODUÇÃO	3
1.1 MOTIVAÇÃO.....	3
1.2 OBJETIVOS GERAIS.....	4
1.3 ESTRUTURAÇÃO DO TRABALHO.....	5
2 CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS	6
2.1 TRABALHOS RELACIONADOS.....	6
2.2 REFERENCIAL TEÓRICO.....	7
2.2.1 <i>Mineração de Padrões Frequentes</i>	7
2.2.2 <i>Suporte e Itemsets Frequentes</i>	7
2.2.3 <i>Algoritmo FP-Growth</i>	8
2.2.4 <i>Regras de Associação</i>	10
2.2.5 <i>Grafos</i>	11
2.2.6 <i>Redes Complexas</i>	12
3 DESENVOLVIMENTO E RESULTADOS	13
3.1 COLETA DE DADOS.....	13
3.2 PREPARAÇÃO DOS DADOS.....	14
3.3 APLICAÇÃO DO ALGORITMO FP-GROWTH.....	16
3.4 APRIMORAMENTO DO ALGORITMO FORMADOR DE EQUIPES.....	20
3.5 MODELAGEM DO GRAFO.....	21
3.6 ANÁLISE EXPLORATÓRIA DA POKÉDEX.....	24
3.7 DESENVOLVIMENTO E IMPLANTAÇÃO DO SISTEMA WEB.....	30
4 CONCLUSÕES (E TRABALHOS FUTUROS)	33
5 REFERÊNCIAS	34

1 INTRODUÇÃO

Pokémon é uma franquia muito popular de jogos desenvolvidos pela *Game Freak* e publicados pela *Nintendo* desde 1996 – quando ocorreu o lançamento de *Pokémon Red & Blue* (GAME FREAK, 2022). Desde então, vários jogos foram lançados com uma periodicidade de 2 em 2 anos, aproximadamente. Assim como os gráficos evoluíram de um 2D, “preto e branco” para um 3D, colorido e moderno como os demais jogos da atualidade, as configurações possíveis para cada pokémon e o efeito delas em batalha vêm se tornando cada vez mais profundas e complexas e o uso dos monstros deixou de ser apenas para vencer as campanhas principais nos jogos e passou a ser uma questão de estratégia para campeonatos reais entre jogadores.

A *Smogon* é um website de Pokémon e uma comunidade especializada na arte da batalha competitiva (SMOGON, 2022). A plataforma possui, além de fóruns de discussão, uma área chamada *Strategy Pokédex*, onde são publicados guias de estratégia para diferentes gerações de jogos e *tiers* (categorias de pokémons) do competitivo, assim como um simulador de batalhas online chamado *Pokémon Showdown*.

Em termos de batalhas competitivas relacionadas a Pokémon, a *Smogon* é o mais recurso online mais abrangente e, além das inúmeras competições menores que ela oferece, o circuito de torneios chamado *Smogon Tour* é uma das peças centrais do ambiente competitivo de Pokémon na internet (SMOGON, 2022).

Uma das categorias mais populares da *Smogon* chama-se *OverUsed* (OU) e é baseada em usagem: pokémons encontrados em mais de 50% das batalhas são considerados dessa *tier*. Nesse contexto, análises acerca das centenas de pokémons existentes nos jogos e estratégias de formação de equipes baseadas em técnicas de mineração de dados são bem-vindas e podem ser valiosas para aqueles que buscam formar boas equipes para competir em *tiers* como essa.

1.1 Motivação

Apesar da *Smogon* contar com uma ferramenta de sugestão de estratégias para cada pokémon existente, muitos jogadores podem não desenvolver boas estratégias por não entenderem o suficiente da arte do competitivo ou se sentirem “perdidos” em meio a tantas opções e detalhes.

Uma forma de facilitar as escolhas desses jogadores a respeito de quais pokémons utilizar em um time é se basear nas escolhas daqueles que entendem sobre o assunto, ou seja,

jogadores assíduos do competitivo. Padrões observados em times frequentemente utilizados por jogadores dedicados podem se tornar boas sugestões para aqueles que estão iniciando no meio. Uma vez que as indecisões relativas a quais pokémons utilizar forem tratadas, a questão de qual estratégia utilizar com cada pokémon pode ser facilmente resolvida com o acervo de informações da *Smogon*.

Com o uso de técnicas de *mineração de padrões frequentes* e construção de *regras de associação* um algoritmo de sugestão de equipes pode ser facilmente desenvolvido e um sistema web que o disponibilize ao público, implantado. Além disso, a fim de sanar dúvidas e descobrir curiosidades sobre os diversos pokémons existentes, o bom proveito de ferramentas capazes de manipular com eficiência o volume de dados disponíveis sobre todos os pokémons e revelar características sobre eles através dos diversos tipos de visualização de dados pode levar a descobertas úteis e interessantes que não seriam possíveis de se alcançar ao analisar um pokémon individualmente.

1.2 Objetivos gerais

O objetivo deste projeto é explorar a base de dados contendo informações sobre todos os pokémons existentes até os jogos da nona geração da franquia principal, respondendo perguntas interessantes sobre os monstros, comparando seus diversos tipos e as gerações dos jogos em busca de curiosidades que algum dia todo jogador pode ter se questionado a respeito, assim como analisar uma amostra de equipes utilizadas em batalhas competitivas na *tier OverUsed* do *Pokémon Showdown* por jogadores assíduos, utilizando técnicas de mineração de padrões frequentes e visualização de relacionamentos como grafos.

Tais tarefas podem ser divididas em: uma análise exploratória de uma base de dados retirada do *Kaggle* contendo dezenas de informações sobre cada pokémon, buscando relacionar cada uma a fim de obter *insights* interessantes; a utilização de *mineração de padrões frequentes* e *regras de associação* para construir novas e boas equipes de pokémons para a *tier OU* baseado em equipes já utilizadas por outros jogadores; o uso de *redes complexas* para visualizar a centralidade de pokémons da amostra extraída do *Pokémon Showdown* e o desenvolvimento de um sistema web que extraia o algoritmo desenvolvido da teoria e o disponibilize de modo que o público geral consiga utilizá-lo.

Essas análises são importantes pelo fato de que, além de trazer visões interessantes sobre esse tema através de novas abordagens, são uma maneira de exemplificar possíveis casos de uso dos algoritmos estudados durante a graduação em contextos populares, tornando mais didático o entendimento de conceitos complexos.

1.3 Estruturação do trabalho

A fim de melhor compreensão desta monografia, foi feita a divisão dela em capítulos. O Capítulo 2 apresenta a contextualização e trabalhos relacionados, em que são descritos todos os termos e elementos necessários para o entendimento da obra, assim como trabalhos correlatos presentes na literatura. O Capítulo 3 apresenta o desenvolvimento do trabalho, com o passo a passo de tudo que foi construído a fim de servir como referência, bem como apresenta os resultados obtidos no decorrer do processo. Por fim, o Capítulo 4 conclui esta obra e apresenta o que poderá ser desenvolvido a fim de aprimorá-la no futuro.

2 CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS

2.1 Trabalhos Relacionados

Através de uma revisão da literatura é possível notar que o tema *Pokémon* é bastante utilizado para explorar conceitos da computação. Scott Lee e Julian Togelius, em *Showdown AI Competition* (LEE; TOGELIUS, 2017) apresentam um jogo baseado em competição de IAs construído em torno de um clone de *Pokémon Showdown*. A. Alfonso (2019) também explora o uso de IAs através de um simulador automático de batalhas desenvolvido por ele e disponível para uso em seu *github*, no qual um usuário forma duas equipes e, a partir disso, o simulador monta os ataques de cada pokémon e simula uma batalha entre eles.

Trabalhos que visam, através de técnicas de aprendizado por reforço, treinar IAs para que se saiam bem contra humanos ou batalhem entre si também são comuns. Scott Lee, dessa vez com Dan Huang, em *A Self-Play Policy Optimization Approach to Battling Pokémon* (HUANG; LEE, 2019) explora uma abordagem de aprendizado por reforço baseada em *self-play* de baixo custo, em que o agente é treinado em diferentes ambientes e se mostra capaz de atuar bem e ser competitivo em relação a jogadores humanos.

Grady Zielke (2020) explora, em sua tese de mestrado, a forma como a *Smogon* afeta a maneira das pessoas jogarem após tomarem conhecimento da mecânica proposta pela comunidade e evidencia que sua influência é inegável, assim como a forma dos jogadores se comportarem nos jogos da franquia principal (RPGs lançados para os videogames da *Nintendo*) e no *Pokémon Showdown* é muito diferente, a partir do momento em que tomam ciência das estratégias e regras propostas. Um segundo exemplo de como a *Smogon* molda a preocupação dos jogadores em relação às estratégias do competitivo é evidenciado pelo interesse de Verdear e Visser (2021) em desenvolver uma ferramenta avaliadora de equipes baseada em ontologia, que dada uma formação de equipe, mapeia os papéis de cada pokémon e fornece uma métrica rápida de viabilidade do time.

O *Pikalytics*, uma plataforma online gratuita que fornece auxílio de diversas maneiras para jogadores que objetivam montar uma equipe, possui um *team builder*, que sugere pokémons e baseado nas escolhas do jogador, sugere os próximos. No entanto, não é divulgado o funcionamento por trás dessa ferramenta. Esta monografia visa explorar o potencial de técnicas de mineração de dados relativamente simples em prover sugestões satisfatórias de formação de equipes baseadas em padrões frequentes encontrados em equipes reais de jogadores do *Pokémon Showdown*.

2.2 Referencial Teórico

Essa subseção tem como objetivo explicar os principais conceitos teóricos necessários para entender o presente trabalho.

2.2.1 Mineração de Padrões Frequentes

Mineração de Padrões Frequentes diz respeito à tarefa de minerar padrões informativos e úteis em conjuntos de dados massivos e complexos (M. J. Zaki, W. Meira, Jr., 2020). Os padrões podem ser minerados de conjuntos de itens, sequências e até grafos. Esta presente obra fará uso da mineração de padrões frequentes em conjuntos de itens, que se baseia na co-ocorrência de valores de atributos ou itens de uma base de dados. A seguir serão explicados alguns termos frequentes:

- **Itemsets:** seja $\mathbf{I} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ um conjunto de elementos chamados itens. Um conjunto $\mathbf{X} \subseteq \mathbf{I}$ é chamado conjunto de itens ou *itemset*. Um conjunto de itens de tamanho k é chamado *k-itemset*.
- **Tidsets:** seja $\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n\}$ outro conjunto de elementos, dessa vez chamados de *tids* ou identificadores de transação. Um conjunto $\mathbf{Y} \subseteq \mathbf{T}$ é chamado *tidset*. *Itemsets* e *tidsets* são mantidos em ordem lexicográfica.
- **Transações:** uma transação é uma tupla da forma $\langle \mathbf{t}, \mathbf{X} \rangle$ tal que $\mathbf{t} \in \mathbf{T}$ é um *tid* único e \mathbf{X} é um *itemset*.
- **Base de dados:** uma base de dados \mathbf{D} pode ser vista como uma relação de conjuntos de *tids* e itens, tal que $\mathbf{D} \subseteq \mathbf{T} \times \mathbf{I}$. Nós dizemos que um *tid* $\mathbf{t} \in \mathbf{T}$ possui o item $\mathbf{x} \in \mathbf{I}$ se, e somente se, $\langle \mathbf{t}, \mathbf{x} \rangle \in \mathbf{D}$.

2.2.2 Suporte e *Itemsets* Frequentes

O *suporte* de um *itemset* \mathbf{X} em uma base de dados \mathbf{D} , chamado de $sup(\mathbf{X})$ nada mais é do que o número de transações em \mathbf{D} que contém \mathbf{X} . O *suporte relativo* de \mathbf{X} é a fração de transações que contém \mathbf{X} : $rsup(\mathbf{X}) = sup(\mathbf{X}) / |\mathbf{D}|$. Um *itemset* é dito frequente se seu suporte é maior ou igual a um suporte mínimo definido pelo usuário.

2.2.3 Algoritmo *FP-Growth*

Existem vários algoritmos para descoberta de *itemsets* frequentes e o *FP-Growth* é um deles, que se destaca pelo seu desempenho. Este algoritmo utiliza uma estrutura de árvore para representar a base de dados, conhecida como *frequent-pattern tree* ou *FP tree*. Inicialmente ocorre uma filtragem nos itens de cada transação – deixando apenas os que possuem suporte maior ou igual ao suporte mínimo – e eles são ordenados de forma decrescente para cada transação. A raiz da árvore possui o valor *null* e cada nó carrega como rótulo, o item, e valor, o suporte do padrão formado pelos itens listados da raiz até o presente nó. Um exemplo de construção da *FP tree*, retirado do livro *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*, de Mohammed J. Zaki e Wagner Meira Jr., é apresentado a seguir. Considere o quadro 1 como uma representação de uma base de dados **D**, cujas transações já estão com os itens ordenados de forma decrescente.

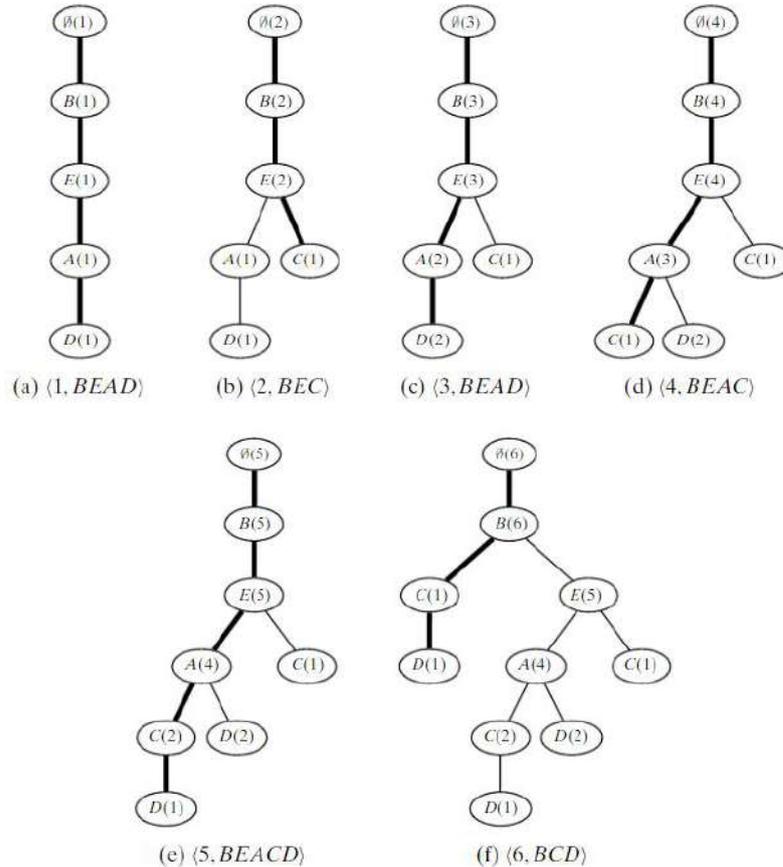
Quadro 1. Base de dados D com itens em ordem decrescente por transação

Transações
BEAD
BEC
BEAD
BEAC
BEACD
BCD

Fonte: M. J. Zaki, W. Meira, Jr.; 2020 (adaptado)

A figura 1 ilustra o processo de inserção e incremento do suporte para cada *itemset*.

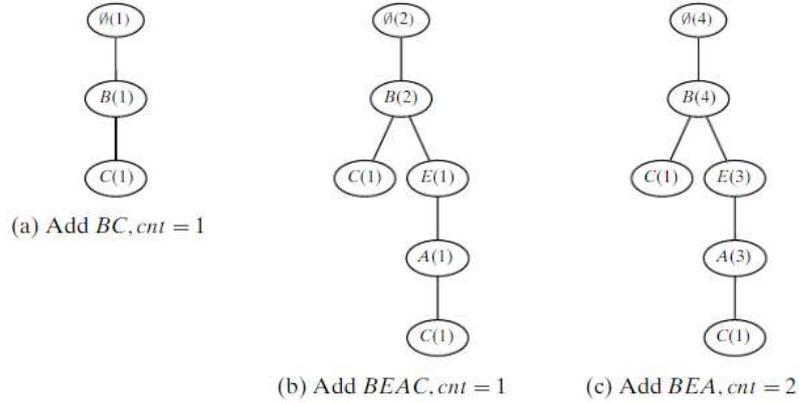
Figura 1 – FP tree.



Fonte: M. J. Zaki, W. Meira, Jr., 2020, p. 234.

Dada uma FP tree \mathbf{R} , árvores projetadas são construídas para cada item frequente \mathbf{i} em \mathbf{R} em ordem crescente de suporte, de maneira recursiva. A fim de projetar \mathbf{R} no item \mathbf{i} , encontra-se todas as ocorrências de \mathbf{i} na árvore e, para cada ocorrência, determina-se o caminho correspondente da raiz até \mathbf{i} . A contagem do item \mathbf{i} em um determinado caminho é registrada em $cnt(\mathbf{i})$ e o caminho é inserido na nova árvore projetada \mathbf{RX} , onde \mathbf{X} é o itemset obtido pela extensão do prefixo \mathbf{P} com o item \mathbf{i} . Ao inserir o caminho, a contagem de cada nó em \mathbf{RX} ao longo do caminho fornecido é incrementada pela contagem de caminho $cnt(\mathbf{i})$. O caso base para a recursão acontece quando a árvore FP de entrada \mathbf{R} é um único caminho. As árvores FP que são caminhos são tratadas pela enumeração de todos os conjuntos de itens que são subconjuntos do caminho, com o suporte de cada um desses conjuntos sendo dado pelo item menos frequente nele. A figura 2 ilustra uma projeção para o item D.

Figura 2 – FP tree projetada para o item **D**.



Fonte: M. J. Zaki, W. Meira, Jr., 2020, p. 234.

A figura 3 apresenta o pseudocódigo do algoritmo. A medida em que as recursões vão ocorrendo, padrões infrequentes são removidos e no final as FP *tree* projetadas para cada item contam apenas com conjuntos de itens frequentes e, ao concatena-los com o respectivo item projetado, temos os *itemsets* frequentes.

Figura 3 – Pseudocódigo do algoritmo *FP-Growth*.

```

// Initial Call:  $R \leftarrow \text{FP-tree}(\mathbf{D})$ ,  $P \leftarrow \emptyset$ ,  $\mathcal{F} \leftarrow \emptyset$ 
FPGROWTH ( $R$ ,  $P$ ,  $\mathcal{F}$ ,  $\text{minsup}$ ):
1 Remove infrequent items from  $R$ 
2 if  $\text{ISPATHTH}(R)$  then // insert subsets of  $R$  into  $\mathcal{F}$ 
3   foreach  $Y \subseteq R$  do
4      $X \leftarrow P \cup Y$ 
5      $\text{sup}(X) \leftarrow \min_{x \in Y} \{\text{cnt}(x)\}$ 
6      $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, \text{sup}(X))\}$ 
7 else // process projected FP-trees for each frequent item  $i$ 
8   foreach  $i \in R$  in increasing order of  $\text{sup}(i)$  do
9      $X \leftarrow P \cup \{i\}$ 
10     $\text{sup}(X) \leftarrow \text{sup}(i)$  // sum of  $\text{cnt}(i)$  for all nodes labeled  $i$ 
11     $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, \text{sup}(X))\}$ 
12     $R_X \leftarrow \emptyset$  // projected FP-tree for  $X$ 
13    foreach  $\text{path} \in \text{PATHFROMROOT}(i)$  do
14       $\text{cnt}(i) \leftarrow \text{count of } i \text{ in path}$ 
15      Insert  $\text{path}$ , excluding  $i$ , into FP-tree  $R_X$  with count  $\text{cnt}(i)$ 
16    if  $R_X \neq \emptyset$  then FPGROWTH ( $R_X$ ,  $X$ ,  $\mathcal{F}$ ,  $\text{minsup}$ )

```

Fonte: M. J. Zaki, W. Meira, Jr., 2020, p. 235.

2.2.4 Regras de Associação

Dada uma coleção de *itemsets* frequentes \mathbf{F} , uma regra de associação é uma expressão da forma $\mathbf{X} \rightarrow \mathbf{Y}$, onde \mathbf{X} e \mathbf{Y} são *itemsets* e eles são disjuntos, ou seja, $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{I}$ e $\mathbf{X} \cap \mathbf{Y} = \emptyset$.

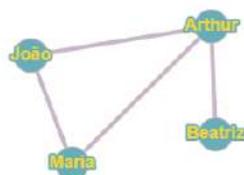
Nesse contexto, **X** é chamado de antecedente e **Y** é chamado de conseqüente. Existem algumas métricas interessantes de se avaliar quando se trata de regras de associação:

- **Suporte:** $sup(\mathbf{X} \rightarrow \mathbf{Y}) = sup(\mathbf{XY}) = sup(\mathbf{X} \cup \mathbf{Y})$.
- **Suporte relativo:** $rsup(\mathbf{X} \rightarrow \mathbf{Y}) = rsup(\mathbf{XY})/|\mathbf{D}| = P(\mathbf{X} \wedge \mathbf{Y})$
- **Confiança:** indica o quão frequente a regra se mostrou válida. É dada pela probabilidade condicional de uma transação conter **Y** dado que contém **X**:
 - $conf(\mathbf{X} \rightarrow \mathbf{Y}) = P(\mathbf{Y} | \mathbf{X}) = P(\mathbf{X} \wedge \mathbf{Y})/P(\mathbf{X}) = sup(\mathbf{XY})/sup(\mathbf{X})$
- **Lift:** expressa a chance de Y ser escolhido, se X for escolhido, considerando toda a popularidade de Y.
 - Um *lift* maior que 1 indica que o item Y é provável de ser escolhido quando X for escolhido.
 - Um *lift* menor ou igual a 1 indica que **não** é provável que Y seja escolhido, caso X seja escolhido.
- **Leverage:** mede a diferença entre a probabilidade conjunta observada e esperada de XY assumindo que X e Y são independentes. Deve ser avaliada junto com o *lift*, já que um mesmo *lift* pode vir junto de diferentes valores de *leverage*.
- **Convicção:** calcula a frequência que X é escolhido e Y não, ou seja, ela está interessada em quando a regra falha. Um valor alto de convicção significa que o conseqüente é muito dependente do antecedente.

2.2.5 Grafos

Um grafo $\mathbf{G}(\mathbf{V}, \mathbf{A})$ é definido pelo par de conjuntos **V** e **A** tal que **V** é um conjunto não vazio de vértices ou nós e **A** é um conjunto de pares ordenados (\mathbf{v}, \mathbf{w}) tal que $\mathbf{v}, \mathbf{w} \in \mathbf{V}$. As arestas podem ser direcionadas (representadas por setas) ou não. Grafos normalmente são utilizados para representar relações entre entidades, por exemplo: sejam os vértices pessoas e as arestas a relação de amizade. O grafo não direcionado da figura 4 mostra que João, Arthur e Maria são amigos e que Beatriz é amiga de Arthur, mas não de João e Maria.

Figura 4 - Grafo não direcionado representando a relação de amizade entre pessoas



Fonte: <https://graphonline.ru/pt/>

Uma medida simples de centralidade em um grafo é o **grau**. O grau de um vértice v é igual ao número de arestas que incidem em v . No exemplo da figura 4, o grau do vértice de João é igual a 2, de Maria é 2, de Arthur é 3 e de Beatriz é 1.

2.2.6 Redes Complexas

Segundo Albert e Barabási (2002), o termo “redes complexas” refere-se a um grafo que apresenta características topológicas não triviais que não ocorrem em grafos puramente aleatórios, como por exemplo alta aglomeração, caminho médio pequeno e falta de escala. Esse conceito é comumente utilizado para representar relações existentes em sistemas complexos, como sistemas sociais, biológicos, tecnológicos e físicos, alguns exemplos são: relação de amizades em uma rede social, redes neurais no cérebro e relação de páginas da web.

3 DESENVOLVIMENTO E RESULTADOS

3.1 Coleta de dados

Os dados foram coletados de quatro fontes diferentes. A fim de realizar a análise exploratória sobre os pokémons da 1ª à 8ª geração foi utilizada uma base de dados retirada do *Kaggle*, contendo as seguintes informações sobre cada pokémon: nº na pokédex, nome, geração, classificação, habilidades, peso e altura, tipo primário e secundário, pontos de ataque, defesa, velocidade, ataque especial e defesa especial, assim como a soma desses atributos, efetividade de outros tipos contra ele, taxa de captura, se é lendário, mítico ou mega evolução, número de passos necessários para chocar um ovo da espécie e valor base de felicidade. O *dataset*, então, foi complementado com uma segunda base, também retirada do *Kaggle*, que possuía menos informações sobre os pokémons da 1ª à 9ª geração, mas era uma boa forma de extrair as informações específicas da 9ª geração. Por fim, os dados faltantes para a nova geração foram completados através de *web scraping* do site *Bulbapedia*.

Para a mineração de padrões frequentes foram utilizadas amostras de equipes reais usadas por jogadores da *tier OU* no simulador online de batalha *Pokémon Showdown*.

Inicialmente, quando a 9ª geração ainda não havia sido lançada, tanto a coleta de dados quanto a aplicação do algoritmo formador de equipes foram baseados apenas na *tier OU gen 8*, cujos processos de obtenção e preparação dos dados serão listados primeiro, seguidos pelo processo atualizado para a *tier OU gen 9*.

A primeira amostra veio de um fórum técnico da *Smogon* e faz parte de um conjunto de bases em formato *txt* utilizado por um membro em sua ferramenta de visualização de equipes utilizadas no simulador de batalhas. Da base citada foram extraídas 4340 equipes de seis pokémons. A fim de complementar a base de equipes anterior, foi feito *web scraping* da *ladder* (ranking de top 500 jogadores do *Pokémon Showdown* em termos de pontos) da *tier OU* para a 8ª geração, extraíndo os nomes dos 500 primeiros jogadores, em seguida os links dos 50 últimos replays desses jogadores e, de cada link de replay, foram extraídos os nomes dos pokémons das equipes do jogador e seu adversário. A ferramenta utilizada para isso foi a biblioteca *beautifulsoup* e foi possível extrair novas 16447 equipes de pokémons. Na coleta pós 9ª geração, os dados originaram-se apenas de *web scraping* da *ladder OU gen 9* do *Pokémon Showdown*, visto que um dos objetivos era utilizar dados atualizados. Dessa segunda coleta, saíram apenas 11501 equipes.

3.2 Preparação dos dados

Os principais desafios de preparação de dados vieram na época em que a 9ª geração não existia e parte dos dados vinham da amostra em *txt* obtida no fórum da Smogon, pois como mostra a figura 5, os dados não vieram em formato tabular e continha mais informações do que o necessário, exigindo – além de uma lógica para identificar os inícios e fins de equipes – a exclusão de apelido, informações de gênero, item, habilidade, EVs (*effort values*) e IVs (*individual values*), natureza, golpes e sobre o pokémon ser ou não brilhante. Também foram filtradas equipes que continham menos de 6 pokémons.

Figura 5 - Amostra retirada do fórum técnico da *Smogon* em formato *txt*

```
gen8ou - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
=== [gen8ou] OverUsed/#20596 78 Likes 63 Score posted by airfare sz stoise digs
webs https://www.smogon.com/forums/threads/ss-ou-metagame-discussion-thread-v7-
usage-stats-in-post-3539.3672210/page-206#post-9309272 ===

Lumberjack (Diggersby) (M) @ Life Orb
Ability: Huge Power
Shiny: Yes
EVs: 4 HP / 252 Atk / 252 Spe
Adamant Nature
- Mega Kick
- Thunder Punch
- Swords Dance
- Quick Attack

Wyatt (Shuckle) (M) @ Mental Herb
Ability: Sturdy
Shiny: Yes
EVs: 248 HP / 136 Def / 124 SpD
Sassy Nature
IVs: 0 Atk
- Stealth Rock
- Sticky Web
- Final Gambit
- Encore

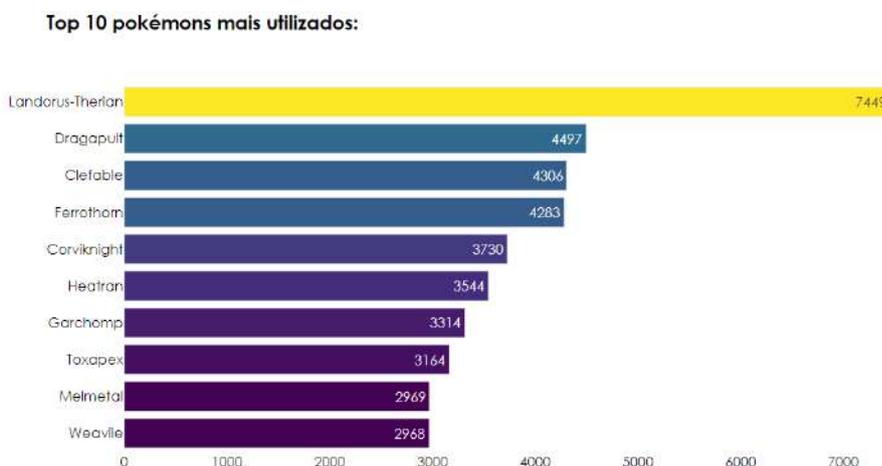
Ln 9, Col 15 90% Unix (LF) UTF-8
Fonte: https://www.smogon.com/forums/threads/smogon-rmt-team-dump.3622884
```

Um *dataframe* pandas foi construído a partir a amostra em *txt* contendo 6 colunas, sendo cada uma relativa a um pokémon e, portanto, cada linha contendo uma equipe utilizada no competitivo. Em seguida, as duas amostras foram combinadas, gerando um *dataframe* contendo 20787 equipes utilizadas na *tier* OU com 578 pokémons distintos. Na segunda versão, considerando a *tier* OU *gen 9*, não foi necessário combinar a base advinda de *web scraping* com nenhuma outra e seu tamanho era de 11501 equipes, possuindo 392 pokémons distintos – valor cuja redução em relação à base pré 9ª geração provavelmente se deve ao fato de muitos pokémons deixarem de ser utilizados pelos melhores jogadores, com a alteração do *metagame* (estilo de jogo).

Da base de dados resultante já foi possível observar a popularidade de alguns pokémons através de um simples gráfico de barras. A figura 6 revela quais foram os 10

pokémons mais populares da amostra utilizada na época pré 9ª geração. O pokémon *Landorus-Therian* se destacou em relação ao segundo e terceiro pokémons mais utilizados (*Dragapult* e *Clefable*), sendo utilizado mais de 7 mil vezes!

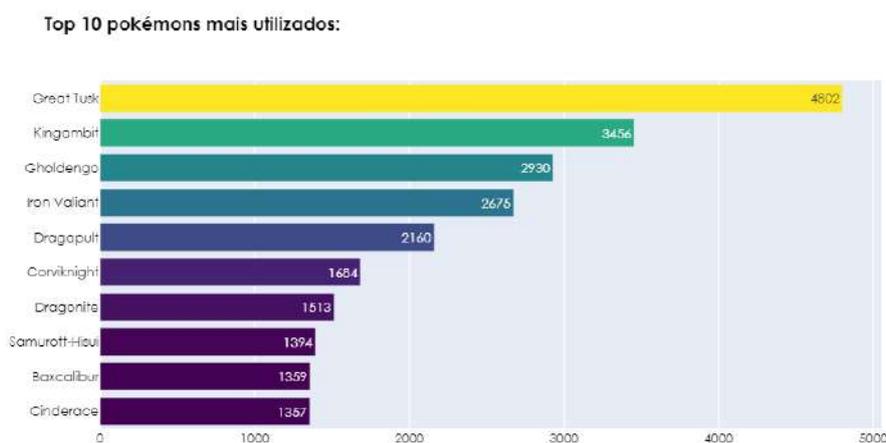
Figura 6 - 10 pokémons mais utilizados (*gen 8* OU)



Fonte: *Jupyter Notebook* (do próprio autor)

A figura 7 apresenta o novo ranking de 10 pokémons mais utilizados com o lançamento da 9ª geração. É possível ver que, dos pokémons que compunham o ranking antigo, restaram apenas *Dragapult* e *Corviknight*, ambos sendo utilizados com uma frequência próxima de 50% do que era antes. Dessa vez quem se destaca no pódio é *Great Tusk*, pokémon da 9ª geração, seguido por *Kingambit* e *Gholdengo*.

Figura 7 - 10 pokémons mais utilizados (*gen 9* OU)



Fonte: *Jupyter Notebook* (do próprio autor)

A fim de preparar a amostra para a aplicação do algoritmo de mineração de padrões frequentes *FP-Growth*, o *dataframe* foi transformado de forma que cada espécie de pokémon

da amostra se tornasse uma coluna e as linhas fossem compostas por valores 0 ou 1, indicando ausência ou presença, respectivamente, dos pokémons nas equipes. Quanto à base de pokémons da 1ª à 9ª geração, para a análise exploratória foi necessário apenas fazer algumas conversões de tipo de atributo e preencher os poucos valores nulos com os valores reais, através de pesquisa na internet.

3.3 Aplicação do Algoritmo FP-Growth

O algoritmo de mineração de padrões frequentes escolhido foi o *FP-Growth* e essa escolha se deu pelo fato de que, considerando que inicialmente se tratavam de quase 21 mil linhas na base, era necessário um algoritmo eficiente para uma quantidade considerável de dados e esse algoritmo possui um desempenho superior a outros algoritmos como *apriori* e *ECLET*.

Após aplicar o algoritmo nos dados já preparados, constatou-se que para um suporte acima de 9,95% (12,34% com a adição da 9ª geração) já não existiam *itemsets* frequentes com mais de um pokémon, fazendo com que fosse necessário tentar valores de suporte bem pequenos, a fim de ter dados suficientes para explorar as regras de associação. Diante disso, o suporte mínimo utilizado foi de 0,1% e as regras de associação foram geradas, com o foco passando a ser nas métricas de confiança e *lift*.

Inicialmente, o *itemset* mais frequente na amostra possuía suporte relativo de 35,82% e tratava-se de *Landorus-Therian*, o que não foi nenhuma surpresa dado o gráfico apresentado anteriormente. Com a adição da 9ª geração, o *itemset* mais frequente passou a ser *Great Tusk*, com suporte relativo de 41,75%.

Após gerar as regras de associação utilizando a função *association_rules*, do pacote *mlxtend.frequent_patterns* e, em seguida, foi construído o algoritmo de formação de equipes, cujo código-fonte – escrito na linguagem *Python* – da primeira versão pode ser encontrado a seguir.

```
team = []
pokemon = ''
nth_pokemon = 1
```

```

while True:
    pokemon = input(f'Insira o nome do {nth_pokemon}º pokémon da sua equipe:
').title()
    nth_pokemon += 1
    team.append(pokemon)
    if is_team_complete(team):
        print("Resultado: " + str(team)[1:-1])
        break
    else:
        suggestions = rules_df[rules_df['antecedents'] == frozenset(team)].copy()
        suggestions = suggestions[suggestions['consequent_len'] == 1]
        suggestions = suggestions[['antecedents', 'consequents', 'confidence',
'lift']]
        suggestions = suggestions.sort_values(by = ['confidence', 'lift'], ascending
= False).reset_index(drop = True)
        suggestions.rename(columns = {'antecedents':'Equipe atual',
'consequents':'Sugestão', 'confidence':'Confiança', 'lift':'Lift'},
inplace = True)
        suggestions['Confiança'] = [str(round(valor * 100, 2)) + '%' for valor in
suggestions['Confiança']]

        if has_suggestions(suggestions):
            print("Pokémons atuais: " + str(team))
            print(f"\nEssas são as sugestões para o {nth_pokemon}º:")
            display(suggestions.head(10))
        else:
            print("Pokémons atuais: " + str(team))
            if pokemon not in unique_pokemons:
                team.pop()
                print('O pokémon escolhido não está disponível ainda, tente outro
nome válido.')
            else:
                print("Oops, não temos mais sugestões para essa equipe :(")
                print("\nResultado: " + str(team)[1:-1])
                break
        nth_pokemon -= 1

```

Nas figuras 8, 9, 10, 11 e 12 é possível visualizar a primeira versão do algoritmo em funcionamento sugerindo pokémons até que seja formada uma equipe de 6 pokémons, assim como as métricas de confiança e *lift* associadas a cada sugestão.

Figura 8 - Sugestões do algoritmo para o segundo pokémon

Insira o nome do 1º pokémon da sua equipe: Heatran
Pokémons atuais: ['Heatran']

Essas são as sugestões para o 2º:

	Equipe atual	Sugestão	Confiança	Lift
0	(Heatran)	(Landorus-Therian)	58.24%	1.625211
1	(Heatran)	(Weavile)	28.05%	1.967673
2	(Heatran)	(Dragapult)	21.81%	1.010015
3	(Heatran)	(Tornadus-Therian)	18.54%	1.453081
4	(Heatran)	(Rotom-Wash)	16.79%	2.158266
5	(Heatran)	(Kartana)	16.59%	1.299985
6	(Heatran)	(Rillaboom)	16.34%	1.471434
7	(Heatran)	(Melmetal)	15.77%	1.108064
8	(Heatran)	(Tangrowth)	14.76%	3.165746
9	(Heatran)	(Tapu Lele)	14.59%	1.225713

Fonte: *Jupyter Notebook* (do próprio autor)

É possível ver que *Landorus-Therian*, *Weavile* e *Dragapult* são as sugestões mais populares, nesse caso pokémon escolhido será o *Weavile*, por possuir o *lift* mais elevado entre os três.

Figura 9 - Sugestões do algoritmo para o terceiro pokémon

Insira o nome do 2º pokémon da sua equipe: Weavile
Pokémons atuais: ['Heatran', 'Weavile']

Essas são as sugestões para o 3º:

	Equipe atual	Sugestão	Confiança	Lift
0	(Weavile, Heatran)	(Landorus-Therian)	80.28%	2.240322
1	(Weavile, Heatran)	(Rotom-Wash)	46.28%	5.949127
2	(Weavile, Heatran)	(Tangrowth)	43.36%	9.301627
3	(Weavile, Heatran)	(Melmetal)	43.16%	3.031920

Fonte: *Jupyter Notebook* (do próprio autor)

Dado que *Heatran* e *Weavile* foram escolhidos, *Landorus-Therian* é o pokémon cuja probabilidade de ser escolhido é a maior, pois possui uma confiança de 80,28% e seu *lift* também é alto (2,24).

Figura 10 - Sugestões do algoritmo para o quarto pokémon

Insira o nome do 3º pokémon da sua equipe: Landorus-Therian
Pokémons atuais: ['Heatran', 'Weavile', 'Landorus-Therian']

Essas são as sugestões para o 4º:

	Equipe atual	Sugestão	Confiança	Lift
0	(Landorus-Therian, Weavile, Heatran)	(Rotom-Wash)	56.14%	7.217004
1	(Weavile, Heatran, Landorus-Therian)	(Tangrowth)	53.38%	11.451826
2	(Landorus-Therian, Weavile, Heatran)	(Melmetal)	53.13%	3.732586

Fonte: *Jupyter Notebook* (do próprio autor)

Após escolher *Landorus-Therian* como 3º pokémon, três novas sugestões são dadas: *Rotom-Wash*, *Tangrowth* e *Melmetal*. Dessa vez o escolhido será *Tangrowth*, pois embora possua confiança um pouco menor que o primeiro, possui um *lift* muito elevado (11,45).

Figura 11 - Sugestões do algoritmo para o 5º pokémon

Insira o nome do 4º pokémon da sua equipe: Tangrowth
Pokémons atuais: ['Heatran', 'Weavile', 'Landorus-Therian', 'Tangrowth']

Essas são as sugestões para o 5º:

	Equipe atual	Sugestão	Confiança	Lift
0	(Landorus-Therian, Weavile, Heatran, Tangrowth)	(Rotom-Wash)	97.65%	12.553520
1	(Landorus-Therian, Weavile, Heatran, Tangrowth)	(Melmetal)	97.42%	6.843612

Fonte: *Jupyter Notebook* (do próprio autor)

Em seguida, dentre as duas sugestões oferecidas, o pokémon escolhido será *Tangrowth*, por possui tanto a confiança quanto o *lift* maior.

Figura 112 - Sugestões do algoritmo para o 6º pokémon

Insira o nome do 5º pokémon da sua equipe: Rotom-Wash
Pokémons atuais: ['Heatran', 'Weavile', 'Landorus-Therian', 'Tangrowth', 'Rotom-Wash']

Essas são as sugestões para o 6º:

	Equipe atual	Sugestão	Confiança	Lift
0	(Heatran, Weavile, Tangrowth, Rotom-Wash, Land...)	(Melmetal)	99.76%	7.008121

Insira o nome do 6º pokémon da sua equipe: Melmetal
Resultado: 'Heatran', 'Weavile', 'Landorus-Therian', 'Tangrowth', 'Rotom-Wash', 'Melmetal'

Fonte: *Jupyter Notebook* (do próprio autor)

Por fim, a única sugestão oferecida foi o *Melmetal* e, portanto, ele foi escolhido. Com esse exemplo foi possível mostrar que o algoritmo é capaz de guiar um jogador na formação de uma equipe de 6 pokémons baseado em equipes utilizadas por outros jogadores na *tier* OU, sem muita experiência necessária.

3.4 Aprimoramento do Algoritmo Formador de Equipes

A primeira versão do algoritmo era funcional, mas muito limitada em termos de variedade de sugestões. Isso se dava pelo fato de que os pokémons sugeridos eram sempre apenas pokémons que foram frequentemente utilizados com o *itemset* formado por todos os pokémons da equipe atual em conjunto, ou seja, se tivéssemos *Heatran* e *Landorus-Therian*, os pokémons sugeridos seriam apenas aqueles que foram frequentemente utilizados em equipes que possuíam essa dupla, ignorando os casos individuais, como equipes em que havia apenas o *Heatran* ou o *Landorus-Therian* além do pokémon sugerido.

A fim de retirar essa limitação, uma nova versão do algoritmo foi desenvolvida. Nela, sugestões válidas são todas as aquelas que contém o pokémon sugerido como subconjunto, expandindo o leque de opções a serem consideradas. Além disso, foi criada uma coluna que diz respeito à efetividade de cada pokémon sugerido em relação aos tipos que são fraquezas da equipe atual, fornecendo um direcionamento extra para a escolha do próximo pokémon dadas as sugestões. O código referente a essa versão atualizada pode ser visto a seguir.

```
team = []
pokemon = ''
nth_pokemon = 1
while True:
    pokemon = input(f"Insira o nome do {nth_pokemon}º pokémon da sua equipe: ")
    .title()
    nth_pokemon += 1
    team.append(pokemon)
    if is_team_complete(team):
        print("Resultado: " + str(team)[1:-1])
        team_weaknesses = get_team_weaknesses(against_types, pokedex, team)
        print("Fraquezas da equipe final: " + str([weakness[8:] for weakness in
team_weaknesses]))
        break
    elif pokemon not in unique_pokemons:
        team.pop()
        print("O pokémon escolhido não está disponível ainda, tente outro nome
válido.")
        print("Pokémons atuais: " + str(team))
        if len(team) > 0:
            display_grouped_suggestions(suggestions)
            nth_pokemon -= 1
        else:
            suggestions = rules_df[rules_df['antecedents']
                .apply(lambda x: x.issubset(frozenset(team)))]
            .copy()
            suggestions =
suggestions[~suggestions['consequents'].isin(suggestions['antecedents'])]
```

```

suggestions = suggestions[suggestions['consequent_len'] == 1]

against_types = interest_columns[1:]
team_weaknesses = get_team_weaknesses(against_types, pokedex, team)
col_name = "É um reforço contra"
suggestions = fill_choice_advantage(team_weaknesses, pokedex, suggestions,
col_name)

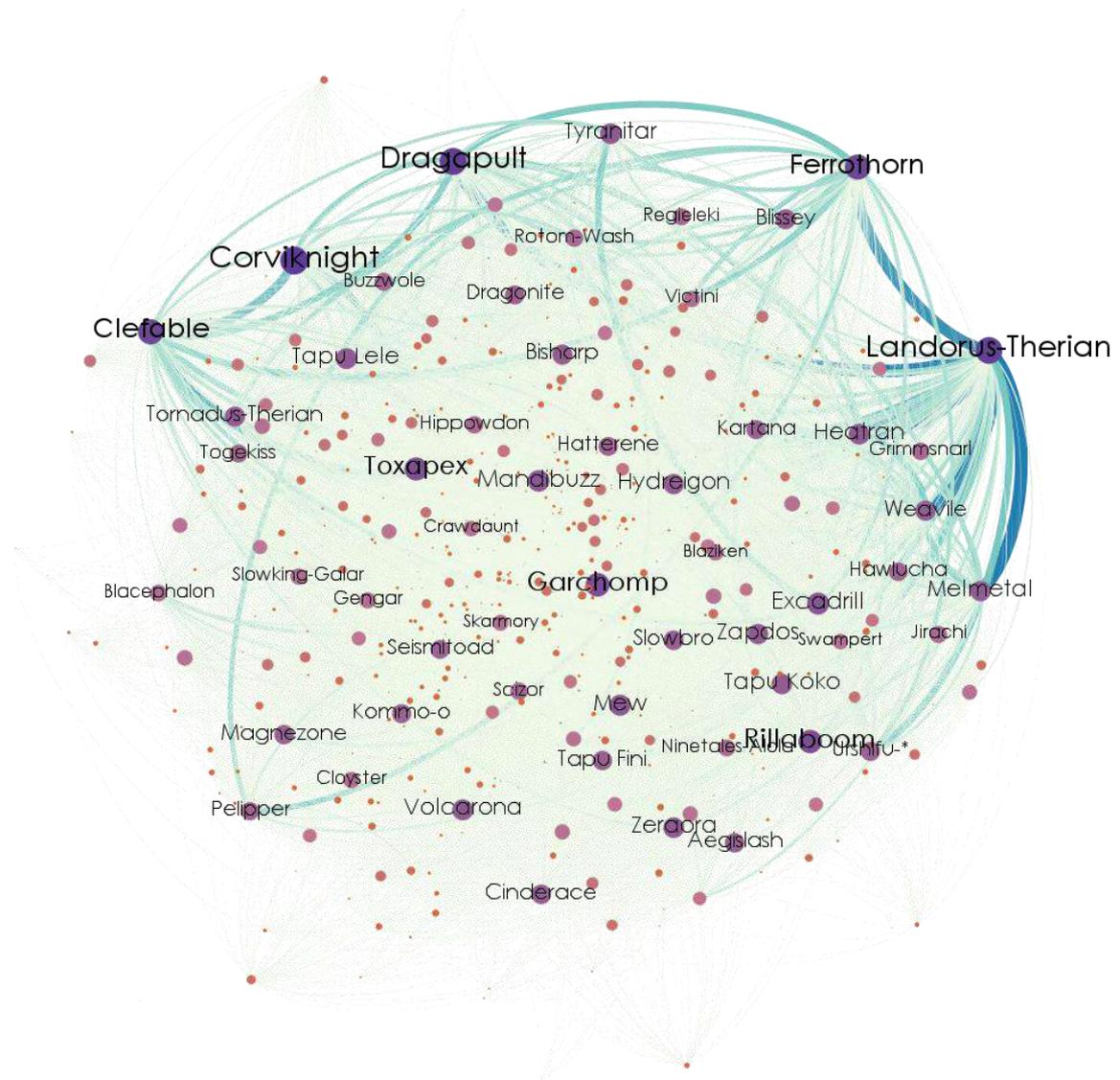
suggestions = suggestions[['antecedents', 'consequents', 'confidence',
'lift', 'É um reforço contra']]
suggestions = suggestions.sort_values(by=['confidence', 'lift'],
ascending=False).reset_index(drop=True)
suggestions.rename(columns = {'antecedents':'Freq. utilizado com',
'consequents':'Sugestão', 'confidence':'Confiança', 'lift':'Lift'}, inplace = True)
suggestions['Confiança'] = [str(round(valor * 100, 2)) + '%' for valor in
suggestions['Confiança']]
if has_suggestions(suggestions):
    print("Pokémons atuais: " + str(team))
    print("Fraquezas da equipe atual: " + str([weakness[8:] for weakness in
team_weaknesses]))
    print(f"\nEssas são as sugestões para o {nth_pokemon}º:")
    display_grouped_suggestions(suggestions)
else:
    print("Oops, não temos mais sugestões para essa equipe :(")
    print("\nResultado: " + str(team)[1:-1])
    nth_pokemon -= 1
    break

```

3.5 Modelagem do Grafo

Os dados da amostra de equipes foram modelados como um grafo, em que os pokémons se tornaram os nós e para cada dois pokémons que apareceram juntos na mesma equipe em algum momento foi criada uma aresta. O grafo foi construído e exportado através da biblioteca *networkx* – da linguagem *Python* – para o software de manipulação e visualização de grafos *Gephi*. Através das figuras 13 e 14 é possível visualizar o resultado para a época pré e pós 9ª geração.

Figura 123 - Visualização do grafo (*gen 8 OU*)

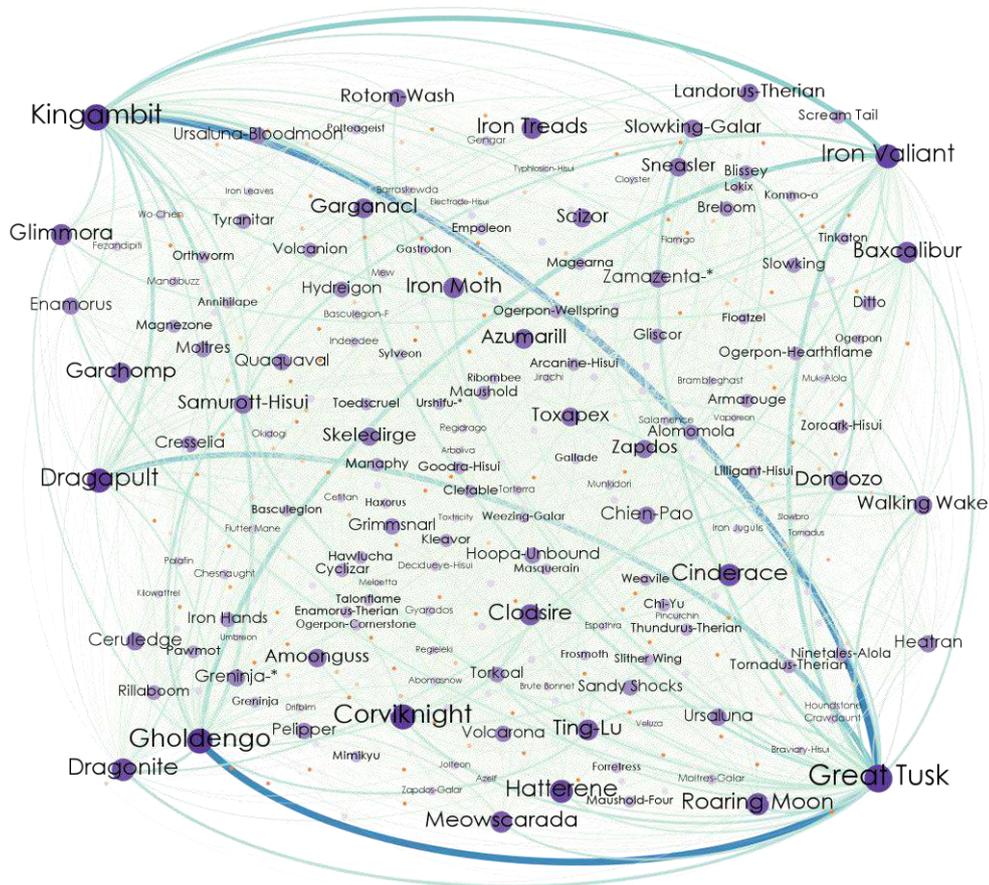


Fonte: Software *Gephi* (do próprio autor)

Como é possível notar, os vértices/nós relativos a *Landorus-Therian*, *Dragapult*, *Clefable*, *Ferrothorn*, *Corviknight*, *Toxapex*, *Garchomp*, *Rillaboom*, *Heatran* e *Melmetal* possuem um tamanho maior que os demais, refletindo a popularidade desses pokémons. Também é possível notar que há arestas de largura considerável saindo do *Landorus-Therian* para o *Ferrothorn*, *Dragapult*, *Melmetal*, dentre outros pokémons, o que significa que esses pokémons são frequentemente utilizados na mesma equipe e que o grau do vértice relativo ao *Landorus-Therian*, por exemplo, também é elevado. O mesmo pode ser dito a respeito da utilização do *Clefable* junto ao *Corviknight* e *Dragapult*, por exemplo. Os vértices/nós cujos nomes dos pokémons são evidenciados são aqueles que tiveram uma recorrência considerável

e é possível notar que são muitos e que também existem muitas arestas conectando diferentes pokémons, revelando uma diversidade de combinações. A versão atualizada do grafo para a *gen 9 OU* pode ser visualizada a seguir.

Figura 134 - Visualização do grafo (*gen 9 OU*)



Fonte: Software *Gephi* (do próprio autor)

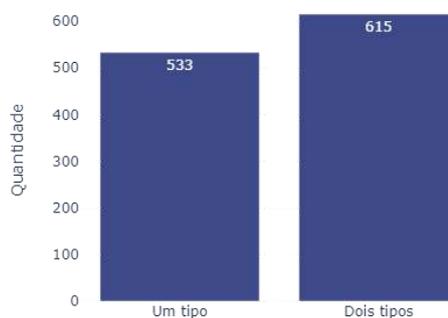
A figura 14 nos permite visualizar o que o gráfico de barras apresentado anteriormente já nos apresentava, além de evidenciar outros pokémons que também são populares, mas não fazem parte do top 10, como *Dragonite*, *Glimmora*, *Landorus-Therian*, *Rotom-Wash*, *Cinderace*, *Donozo*, *Walking Wake*, *Clodsire*, *Garganacl*, *Baxcalibur*, *Iron Treads* e outros. Além disso, é possível ver que alguns dos pokémons mais populares, como *Great Tusk*, *Gholdengo* e *Kingambit* possuem arestas muito espessas entre si, mostrando que são muito utilizados em conjunto.

3.6 Análise Exploratória da Pokédex

Foi realizada uma análise exploratória de dados em cima da base dos pokémons da 1ª a 9ª geração, a fim de responder algumas perguntas e descobrir, no processo, algumas curiosidades. As perguntas podem ser vistas a seguir.

- Como é a proporção de pokémons que possuem apenas um tipo x aqueles que possuem dois tipos?

Figura 15 - Pokémons com 1 tipo x pokémons com 2 tipos

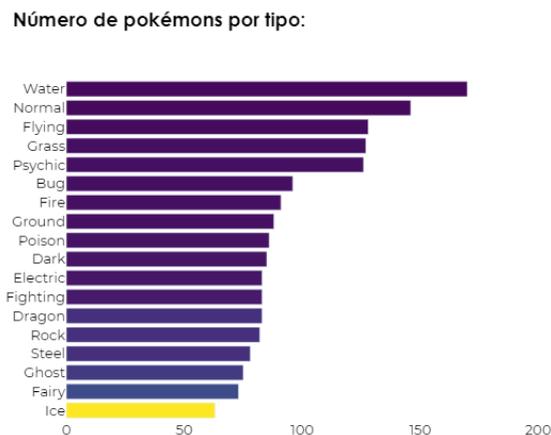


Fonte: *Jupyter Notebook* (do próprio autor)

Como podemos notar, existem mais pokémons com dois tipos do que com apenas um, mas a proporção é relativamente equilibrada, sendo 46,4% pokémons que possuem apenas um tipo e 53,6% que possuem dois (aumento de 0,7% em relação quando não existia 9ª geração).

- Existem quantos pokémons por tipo?

Figura 146 - Número de pokémons por tipo

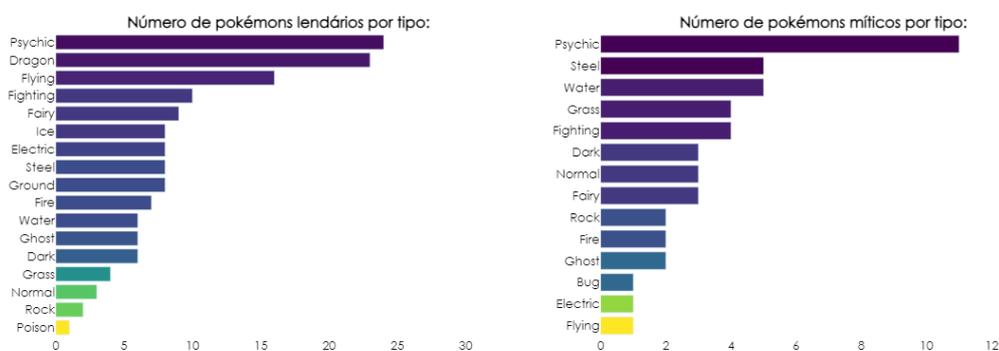


Fonte: *Jupyter Notebook* (do próprio autor)

Através da figura 16 é possível concluir que o tipo **água** é o que possui mais pokémons (com 170), seguido pelos tipos **normal, voador, grama e psíquico** (tendo entre 100 e 150). Por outro lado, vale destacar que o tipo **gelo** ainda possui poucos pokémons (pouco mais de 50). O tipo **fada** (2º tipo com menos pokémons) foi introduzido bem tarde nos jogos (6ª geração), então é compreensível que haja menos pokémons. No entanto, os tipos **fantasma** e **gelo** existem desde a 1ª geração, o que pode indicar que poucos pokémons desses tipos são descobertos a cada geração.

- Qual a proporção de lendários e míticos por tipo?

Figura 157 - Número de pokémons lendários/míticos por tipo



Fonte: *Jupyter Notebook* (do próprio autor)

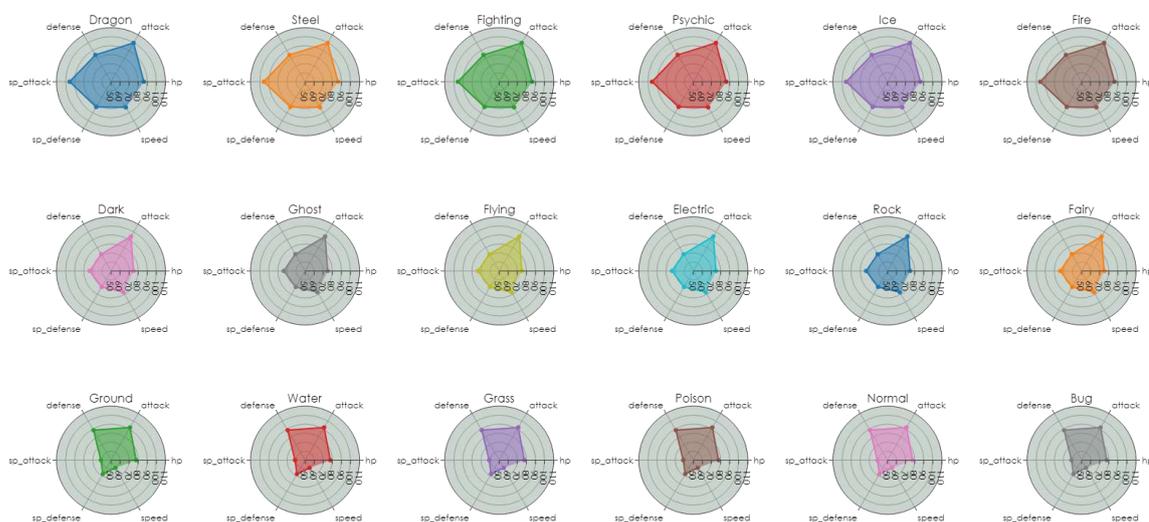
É interessante ver que o tipo **psíquico** predomina tanto entre os pokémons lendários quanto entre os míticos e que no caso dos míticos, a diferença entre a quantidade de pokémons **psíquicos** e o restante é considerável, enquanto no caso dos lendários, os tipos **dragão** e **voador** representam uma parcela também considerável desse grupo. Uma outra observação pertinente é que não existe pokémon lendário do tipo **inseto** e não existe pokémon **mítico** do tipo venenoso ainda.

- Quais as combinações de tipos mais comuns?

Para responder essa pergunta foi gerada uma nuvem de palavras (figura 18) a fim de tornar a visualização de dezenas de combinações mais coerente, de acordo com a popularidade de cada uma.

O ranking de força por tipo foi construído levando em consideração a média da soma dos atributos bases (ataque, defesa, pontos de vida, velocidade, ataque especial e defesa especial) de cada pokémon e mostra que o tipo **dragão** possui, em média, os pokémons mais fortes, enquanto o tipo **inseto** possui, em média, os pokémons mais fracos. Tal diferença de força entre os dois extremos pode ser evidenciada ainda mais através dos gráficos apresentados na figura 20 (ordenados do tipo mais forte para o mais fraco), que apresentam uma visão mais detalhada de cada atributo e revela que a área coberta pelo tipo **inseto** é, de fato, bem menor que aquela que é coberta pelo tipo **dragão**.

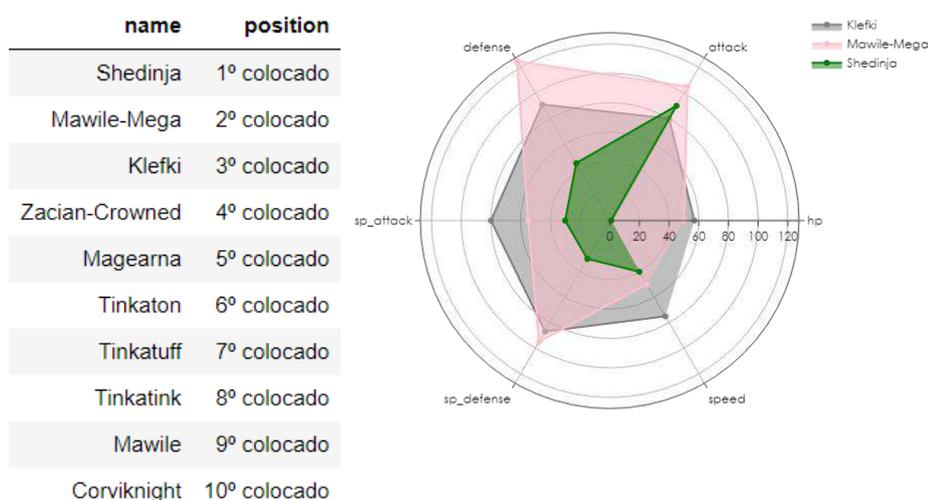
Figura 20 - Visão detalhada de força média por tipo



Fonte: *Jupyter Notebook* (do próprio autor)

- Quais os 10 pokémons mais resistentes a uma maior variedade de tipos?

Figura 181 - Pokémons mais resistentes a uma variedade de tipos



Fonte: *Jupyter Notebook* (do próprio autor)

A figura 21 apresenta quem são os 10 pokémons mais resistentes a uma maior variedade de tipos e compara os 3 primeiros colocados. Através do gráfico é possível ver que, embora o *Shedinja* seja o pokémon resistente à maior variedade de tipos, os valores de seus atributos base são significativamente menores que os dos outros pokémons, o que mostra que talvez ele não seja uma boa escolha só por conta de sua versatilidade em termos de resistência a tipos. No entanto, vale mencionar que o *Shedinja* conta com uma habilidade chamada *Wonder Guard*, que garante que ele só receba dano através de golpes super-efetivos, *Entry Hazards* (golpes como *Spikes* e *Stealth Rock*), efeitos climáticos (*Sandstorm*, *Hail*) ou status (*Burn*, *Poison*, *Badly Poison*), o que compensa um pouco o fato dele possuir o menor *HP* dentre todos os pokémons. Portanto, o segredo é analisar vários fatores ao pensar em escolhê-lo.

No que diz respeito à soma dos atributos base, tem-se que os 3 pokémons mais fortes são três mega-evoluções: *Mega Rayquaza*, *Mega Mewtwo X* e *Mega Mewtwo Y*, que de fato possuem uma soma muito elevada (780).

- Qual geração possui, em média, os pokémons mais fortes? E os mais fracos?

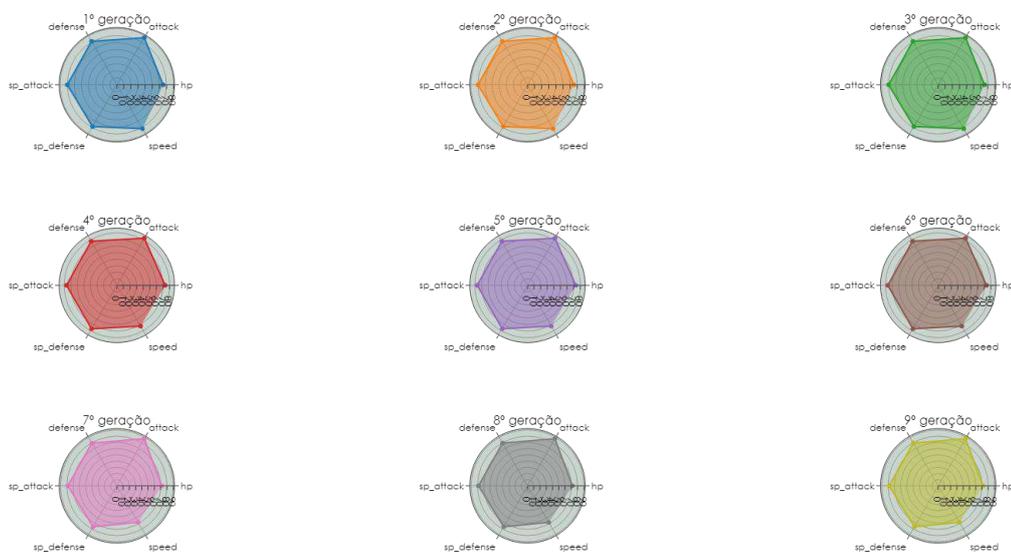
Figura 192 - Ranking de força por geração



Fonte: *Jupyter Notebook* (do próprio autor)

Através da figura 22, é possível ver que a geração que possui os pokémons mais fortes em termos de atributos base é a 7ª, enquanto a detentora dos mais fracos é a 2ª. No entanto, nota-se que as gerações não possuem médias de força muito discrepantes umas das outras, já que em toda geração é possível encontrar tanto pokémons fracos quanto fortes. Os gráficos da figura 23 complementam bem essa constatação, já que revela que a área coberta não é muito diferente em todas as gerações.

Figura 203 - Visão detalhada da força por geração



Fonte: *Jupyter Notebook* (do próprio autor)

3.7 Desenvolvimento e Implantação do Sistema Web

A fim de disponibilizar o algoritmo desenvolvido ao público geral, surgiu a ideia de desenvolver e implantar um sistema web. A ideia é que fosse algo simples, que disponibilizasse ao usuário seis *slots* vazios e a possibilidade do usuário pesquisar por pokémons e preencher esses *slots* com os pokémons sugeridos pelo algoritmo, que seriam apresentados na forma de uma tabela. Além disso, haveria uma seção de *FAQ* (perguntas frequentes).

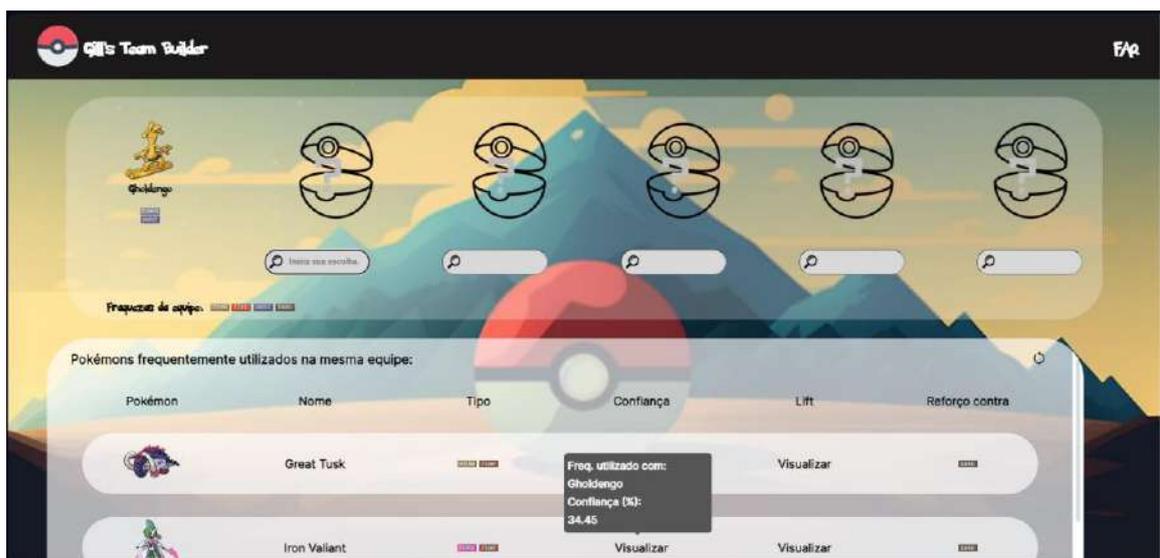
Inicialmente um protótipo de como seriam as telas foi desenvolvido através da ferramenta *Figma* e, com base nele, a primeira versão do *Front-end* foi implementada utilizando *HTML* e *CSS*. Logo em seguida, foi feita a migração do código que estava em cada *jupyter notebook* para scripts *.py*, a fim de executá-los via terminal quando precisasse fazer uma nova coleta via *web scraping*, além da geração e atualização da base de regras de associação, por exemplo, assim como foi desenvolvida uma *API* em *Flask*, que disponibiliza *endpoints* para obter sugestões para a equipe atual de pokémons, seus *sprites* (imagens dos pokémons) e as fraquezas da equipe atual.

Com a *API* implementada, iniciou-se a implementação do código em *JavaScript* responsável por fazer requisições aos *endpoints* mencionados e disponibilizar esses dados ao usuário da interface, bem como adicionar dinamismo ao site. Os principais desafios durante a etapa de desenvolvimento do sistema web se concentraram na parte referente ao *JavaScript* e as mudanças de comportamento da página após pesquisar por sugestões, visto que a página se alterava de forma significativa, além do *CSS*, que mudou bastante de sua primeira versão para a versão final, já que inicialmente as propriedades eram majoritariamente absolutas e, no final, passaram a ser relativas para lidar bem com o dinamismo do site e aspectos básicos de responsividade.

Por fim, tanto a *API* quanto o *Front-End* foram implantados no *pythonanywhere*, cuja categoria *free* concede o direito à implantação de uma aplicação web gratuita. O *Team Builder* está disponível para uso através do link <https://gilliardrodrigues.pythonanywhere.com>. Além disso, inicialmente surgiu-se a ideia de implantar os *scripts* relativos à coleta via *web scraping* e a geração da base de regras de associação na *Google Cloud Platform*, através de *Cloud Functions*, que nada mais é do que uma opção *serverless* cuja *free tier* seria suficiente para este projeto e não geraria custos. O código que efetua *web scraping* foi adaptado para que ficasse dentro do tempo máximo que uma *cloud function* da *GCP* executa e isso se traduziu na redução do tempo de execução de cerca de 1h30 para cerca de 10 minutos, utilizando programação concorrente através da biblioteca *asyncio* da linguagem *Python*. No entanto, próximo da publicação dessa obra, o site do *Pokémon Showdown* foi atualizado para utilizar *JavaScript* e

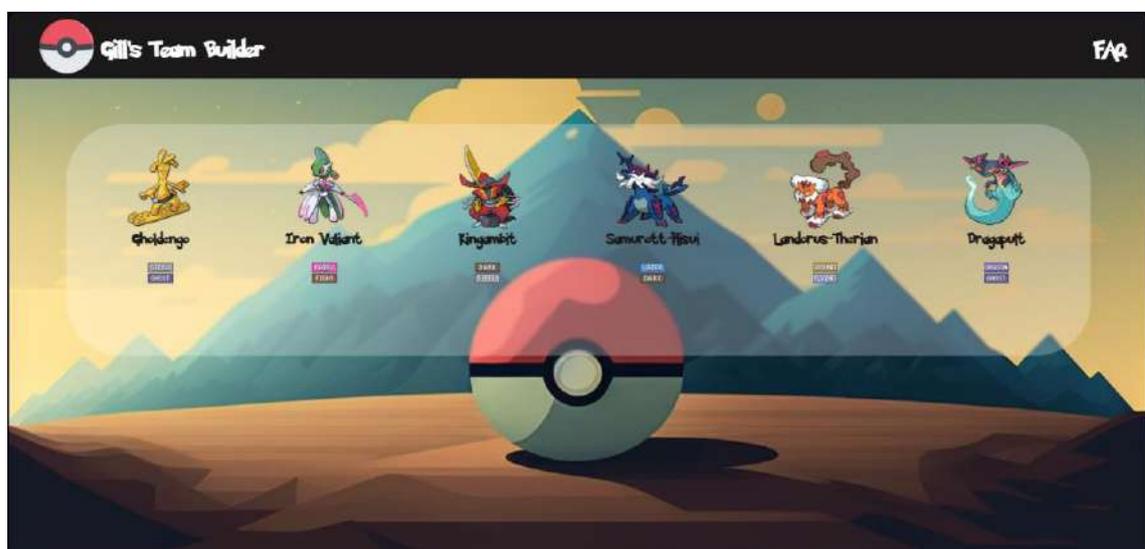
agora o processo original de *web scraping* parou de funcionar, o que inviabilizou a sua implantação. A figura 24 apresenta o resultado final, para o caso em que o usuário escolheu um pokémon e recebeu as sugestões para o próximo, enquanto a figura 25 mostra como fica o site após todos os pokémons terem sido escolhidos e as fraquezas da equipe, neutralizadas. Por fim, na figura 26 é apresentada a tela que possui o *FAQ*.

Figura 214 - Sistema web com as sugestões apresentadas ao usuário



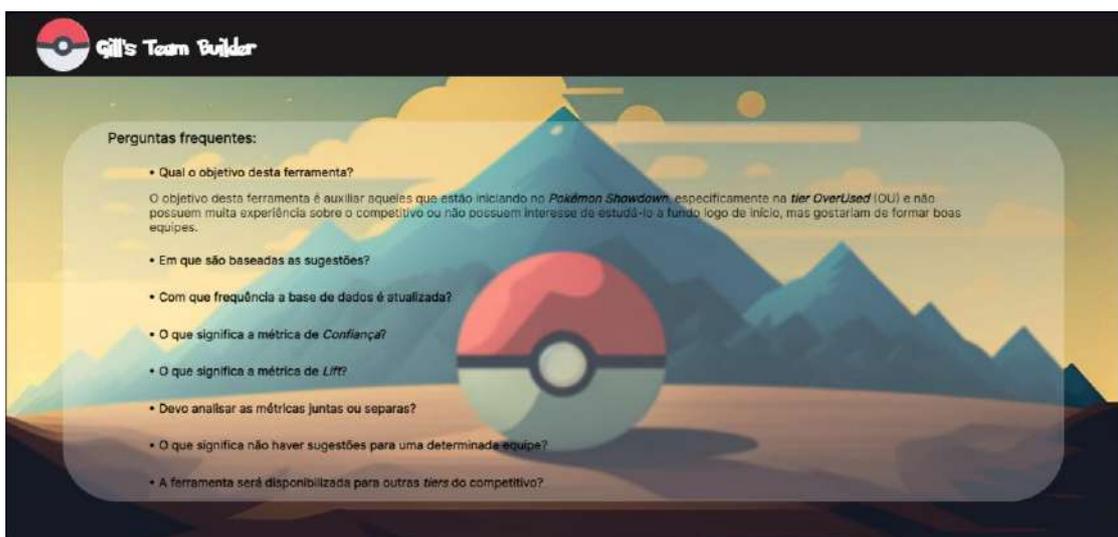
Fonte: Sistema web implantado (do próprio autor)

Figura 225 - Sistema web com a equipe final formada



Fonte: Sistema web implantado (do próprio autor)

Figura 236 - Sistema web com a seção de FAQ à mostra



Fonte: *Sistema web implantado* (do próprio autor)

4 CONCLUSÕES (E TRABALHOS FUTUROS)

Conforme apresentado no início desta obra, os objetivos aqui eram quatro: desenvolver um algoritmo capaz de construir equipes de seis pokémons utilizando sugestões advindas da aplicação de mineração de padrões frequentes, visualizar os relacionamentos entre os pokémons da amostra de equipes utilizadas no *Pokémon Showdown* por jogadores reais através de redes complexas/grafos, fazer uma análise exploratória dos dados dos pokémons da 1ª à 9ª geração e, por fim, implantar um sistema web que disponibilizasse ao público geral o algoritmo formador de equipes desenvolvido.

Baseado nisso, o algoritmo de formação de equipes se mostrou capaz de formar uma equipe de 6 pokémons, alcançando o primeiro objetivo. No entanto, houve contratemplos desagradáveis na primeira versão: a necessidade de um suporte mínimo extremamente baixo e a limitação de algumas combinações não resultarem em equipes de 6 pokémons por falta de sugestões. Diante disso, uma nova versão foi desenvolvida de modo que o problema de variedade de sugestões fosse solucionado. A visualização da amostra de equipes como grafos destacou alguns relacionamentos fortes entre vários pokémons, assim como a popularidade deles separadamente. Tendo isso em mente, pode-se considerar que utilizar essa estrutura de dados para representar a amostra foi uma ideia interessante e valeu a pena.

Ademais, a análise exploratória de dados sobre os pokémons da 1ª à 9ª geração se mostrou capaz de minerar curiosidades bem interessantes sobre os pokémons, como a relação de força por tipo e geração, as combinações de tipo mais comuns, a relação de resistência a variados tipos e força, assim como outras informações relativas à distribuição dos pokémons por tipos. A maioria das questões levantadas levaram a curiosidades que seriam difíceis de descobrir sem uma análise exploratória, reforçando o potencial de bons dados para revelar informações de valor sobre qualquer tema. Por fim, os códigos-fonte relativos a cada etapa do presente trabalho podem ser encontrados no seguinte [repositório do GitHub](#).

Por fim, o algoritmo formador de equipes desenvolvido foi, de fato, implantado e disponibilizado ao público geral através do link <https://gilliardrodrigues.pythonanywhere.com>, o que representou a transição completa da teoria para a prática baseado no objetivo inicial desta obra. Como tema para trabalhos futuros, pode-se listar a atualização do código-fonte relativo à coleta dos dados do *Pokémon Showdown* via *web scraping* com o uso da biblioteca *Selenium* para contornar a adição de *JavaScript* ao site, bem como a adição de sugestões para outras categorias do jogo.

5 REFERÊNCIAS

Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, “*Exploring network structure, dynamics, and function using NetworkX*”, in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6023: informação e documentação: referências: elaboração*. Rio de Janeiro, 2002. 24 p.

Bastian M., Heymann S., Jacomy M. (2009). *Gephi: an open source software for exploring and manipulating networks*. International AAAI Conference on Weblogs and Social Media.

BETETTA, Joshua. *Complete Pokedex (V1.10)*. 2022. Disponível em <<https://www.kaggle.com/datasets/joshuabetetta/complete-pokedex-v100>>. Acesso em 22 set. 2022.

COSTA, Belkiz Inez Rezende; FERNANDES, Wesley Rodrigo. *Normas para publicação de monografias, dissertações e teses para o curso de pós-graduação departamento de Ciência da Computação UFMG*. Disponível em <www.dcc.ufmg.br/especializacao/cei/EngSoft/manualMonografiaDisertTeses.pdf>. Acesso em 10 dez. 2022.

GAME FREAK. 2022. Disponível em <https://pt.wikipedia.org/wiki/Game_Freak>. Acesso em 22 set. 2022.

HUANG, Dan; LEE, Scott. *A Self-Play Policy Optimization Approach to Battling Pokémon*. 2019. IEEE Conference on Games (CoG), 2019, pp. 1-4, doi: 10.1109/CIG.2019.8848014.

LEE, Scott; TOGELIUS, Julian, *Showdown AI competition*. 2017. IEEE Conference on Computational Intelligence and Games (CIG), 2017, pp. 191-198, doi: 10.1109/CIG.2017.8080435.

Mohammed J. Zaki, Wagner Meira, Jr., *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*, 2nd Edition, Cambridge University Press, March 2020. ISBN: 978-1108473989.

ANDRADE, Nazareno. *11 – Visualização de redes com o Gephi*. YouTube, 15 nov. 2020. Disponível em: https://youtu.be/P_NyQC30Z_c. Acesso em 1 nov. 2022.

PIKALYTICS. *Pokemon Team Builder GEN8Gen 7 OverUsed | Pikalytics*. Disponível em <<https://www.pikalytics.com/team/gen8ou>>. Acesso em 22 set. 2022.

Plotly Technologies Inc. Collaborative data science. Montréal, QC, 2015. <https://plot.ly>.

R. Albert, A. L. Barabasi. *Statistical mechanics of complex networks*. *Reviews of Modern Physics* 74, 47 (2002).

Raschka, Sebastian (2018) *MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack*. J Open Source Softw 3(24).

SMOGON. *Smogon University*. 2022. Disponível em www.smogon.com/dex/ss/formats/ou/. Acesso em 22 set. 2022.

SMOGON. *Smogon / RMT Team Dump*. 2022. Disponível em <https://www.smogon.com/forums/threads/smogon-rmt-team-dump.3622884/>. Acesso em 22 set. 2022.

SMOGON. *About Smogon*. Disponível em <https://www.smogon.com/about>. Acesso em 22 set. 2022.

VERDEAR, D.; VISSER, U. *Ontology-based Knowledge System and Team Verification Tool for Competitive Pokemon*. The International FLAIRS Conference Proceedings, [S. l.], v. 34, 2021. DOI: 10.32473/flairs.v34i1.128544. Disponível em: <https://journals.flvc.org/FLAIRS/article/view/128544>. Acesso em: 22 sep. 2022.

ZIELKE, Grady. *Gotta Win'em All: How Expert Play in the Online Community of Smogon Changes Pokemon*. 2020. Dissertação (mestrado) - Department of Digital Humanities, University of Alberta, Edmonton, 2020.