

Universidade Federal do Rio Grande do Sul  
Instituto de Informática

Classificação e Pesquisa de Dados - INF01124

Equipe not(Rocket)

Bibiana Duarte  
Tiago Lucas Flach

Trabalho Final

12 de maio de 2021  
Porto Alegre, RS

## Visão Geral

O trabalho consiste na implementação de um programa que, após extrair os dados de um arquivo, armazena estes dados em arquivos binários e realiza operações de busca e ordenação. Selecionamos como conjunto de dados o seguinte dataset: <https://www.kaggle.com/cristobalmitchell/pokedex>, do qual utilizamos apenas os campos `national_number`, `english_name`, `height_m`, `weight_kg`, `hp`, `attack` e `defense`.

## Implementação

Para o armazenamento dos dados, criamos uma classe, a `Pokemon_Class`, que tem como atributos `num`, `nome`, `hp`, `atk`, `deff`, `peso`, `altura`, e uma árvore Trie, que armazena o nome e o número (`num`).

A aplicação foi desenvolvida em na linguagem Python e utiliza algumas bibliotecas, são elas:

- `csv`
- `os.path`
- `pickle`

Ao executar o código (`main.py`), é feita uma varredura no arquivo `pokemon.csv`, de onde são obtidos cerca de 900 pokemons com seus atributos. Através desta varredura, cada pokemon é transformado em um objeto (`Pokemon_Class`), serializado e armazenado no arquivo `pokemons.pkl` através da biblioteca `pickle`.

Em seguida é criada uma árvore Trie, armazenando nome e número, com os objetos extraídos do arquivo `pickle`. Através desta árvore é possível realizar uma busca por caracteres baseados no nome de cada pokemon, de forma eficiente

## Guia de Uso

A aplicação é utilizada da seguinte forma:

1. Ao executar o arquivo `main.py` será exibida a seguinte mensagem “Pesquise por um Pokemon:”
  - a. Ao digitar, serão exibidas as informações dos pokemons que tiverem ocorrências desta palavra no banco de dados (relacionadas ao nome).

## Contribuições

A aplicação apresenta um mecanismo funcional de busca pelo nome de Pokemon sendo mais eficiente que uma busca linear devido ao uso da árvore Trie.

Porém, pelo fato da aplicação não estar em completo funcionamento, este não apresenta grandes ganhos em uso de memória, sendo que não foi implementado da forma indicada.

## Considerações Finais

Ao avaliar este projeto, podemos observar que o objetivo principal não foi atendido devido a problemas na implementação de código.

A ideia inicial era criar um arquivo pickle com os pokemons armazenados um a um, e após isto criar uma árvore B+ com os índices referenciando os pokemons, para que a árvore B+ pudesse ser ordenada conforme fosse necessário. Porém, não foi possível fazer deste modo, pois não ao extrair objetos de um arquivo pickle, o mesmo é extraído um a um, logo, teria que ser feita uma varredura completa na lista de objetos para encontrar o objeto desejado, o que seria bastante custoso em processamento.

Tendo isto como base, foi iniciada uma tentativa de criar uma árvore B+ e a serializar por completa em um arquivo pickle, o que é contra o objetivo do trabalho, pois estaríamos carregando todos os dados na memória. Porém, não conseguimos implementar a árvore B+ da forma desejada e assim, optamos por deixar esta parte do código não funcional.

## Referências

<https://docs.python.org/3/library/csv.html>

<https://docs.python.org/3/library/os.path.html>

<https://docs.python.org/3/library/pickle.html>

<https://www.kaggle.com/cristobalmitchell/pokedex>

<https://www.javatpoint.com/b-tree-vs-bplus-tree>

<https://www.geeksforgeeks.org/trie-insert-and-search/>