



**ANDREY MESSIAS DE LIMA
CAIO TEIXEIRA LADEIRA
JOÃO GABRIEL DA SILVA NASCIMENTO**

**RELATÓRIO TÉCNICO DO CÓDIGO "CARTAS MONSTROS
YU-GI-OH"**

LAVRAS - MG

1. Introdução

O programa tem a temática YU-GI-OH, focado em cartas do tipo monstro, voltado para a gestão de cartas do jogo. Ele possibilita as seguintes funcionalidades: adicionar, remover, ordenar, e salvar qualquer tipo de alteração feita sobre as cartas. O sistema possui dados dinâmicos proporcionando uma experiência interativa pelo terminal.

2. Estrutura do Código

2.1 Bibliotecas Utilizadas

- `iostream`
- `fstream`
- `cstring`
- `sstream`

2.2 Estruturas Principais

- **`struct carta`**: Representa uma carta de YU-GI-OH com as seguintes variáveis:
 - `nome`, `atributo`, `raridade`, `tipo` (string de tamanho fixo);
 - `nível`, `atk`, `def` (inteiro);
 - `descrição` (string de tamanho fixo).

3. Descrição das Funções

3.1 Funções de Exibição

- **`exibirImagem()`**: Exibe uma arte ASCII no terminal.
- **`tituloMenu()`**: Exibe o título do menu principal.

3.2 Funções de Manipulação de Cartas

- **`prencherVetor()`**: Lê os dados de um arquivo csv e popula um vetor dinâmico de cartas. Possui redimensionamento de vetor caso o vetor atinja sua capacidade máxima.
- **`imprimirVetor()`**: Exibe as cartas em um intervalo desejado.
- **`adicionarCarta()`**: Permite que o usuário adicione uma carta.
- **`removerCarta()`**: Remove uma carta específica com base no nome informado pelo usuário.

3.3 Funções de Ordenação

- **`quickSortInt()` e `quickSortChar()`**: Implementam o algoritmo quicksort para ordenar o vetor de cartas com base em atributos inteiros como `atk`, `nível`, e `def` ou por string como `nome`.

3.4 Funções de Busca

- **`buscaBinariaChar()`**: Realiza uma busca binária por nome.

- **buscaInt()**: Realiza uma busca sequencial para encontrar cartas pela variável nível.
- Obs.: Para buscar dados que se repetem, a lógica da busca binária não se aplica. Por esse motivo, a busca por nível foi feita de forma sequencial

3.5 Funções de Persistência

- **salvarBinario()**: Salva os dados das cartas em arquivo binário.
- **salvarCSV()**: Salva os dados das cartas em um arquivo CSV.

4. Problemas durante a criação do código

O mais notório desafio encontrado no código foi que, devido ao fato que não sabíamos que strings não eram convertidas corretamente para arquivos em binário, a maior parte da lógica do código utilizava strings, e, por isso, tivemos que fazer a substituição para vetor de caracteres, adaptando o código no processo.

Outra dificuldade que tivemos foi que, na impressão correta da imagem em ASCII (uma imagem feita apenas por caracteres de teclado), estavam ocorrendo erros ligados ao sistema operacional da máquina (não funcionava corretamente em windows), que não suportava o ASCII. Porém, pelo trabalho ser avaliado na sua performance em linux, o grupo preferiu não modificar a imagem para que sua formatação funcionasse em todos os sistemas operacionais, visto que a mesma perderia qualidade.

O grupo também teve dificuldades quanto a busca binária, uma vez que a mesma tende a ter problemas quanto a valores repetidos, na maioria das vezes, buscando apenas o primeiro dos valores iguais, ocasionando problemas quanto a busca em um campo com números repetidos. Para lidar com o problema, optamos por utilizar uma busca sequencial para retornar múltiplos números quanto a característica que estava gerando problemas quanto a repetição.

Um desafio encontrado também durante o projeto foi que o grupo tentou fazer a ordenação dos vetores antes que nós tivéssemos as aulas necessárias sobre o tema, assim, tendo que recorrer a internet para encontrar meios de ordenação e adaptando-os para o código de nossa autoria.

5. Conclusão

Esse projeto fez com que todos do grupo desenvolvessem seus conhecimentos sobre o funcionamento de códigos, armazenamento de dados, conversão de arquivos e também acreditamos que esse trabalho nos ajudou a aprimorar a habilidade de planejar códigos de maior complexidade e com maior clareza, pois, devido ao tamanho do algoritmo que deveria ser escrito, seria um desafio muito maior caso não houvesse fases de planejamento e organização do trabalho.

É notório que as habilidades de trabalho em equipe e cooperação também foram treinadas, pois foi a nossa primeira vez tendo que lidar com um código em grupo ao invés de individualmente, assim, todos os membros tiveram que se ajudar e participar do projeto para que tudo ocorresse devidamente.

Em suma, esse projeto foi de extrema utilidade no aprendizado dos membros do grupo sobre a programação, assim como nos ajudou a fortalecer aspectos de cooperação e comunicação.