

Trabalho de Introdução aos Algoritmos (GCC224)

Sistema de cadastro de jogos

Lucas Antônio Lopes Neves

Luiz Carlos Coelho Conde

Thiago Luigi Gonçalves Lima

(Ciência da Computação)

Lavras, 27 de Janeiro de 2018

**Sumário**

1. Introdução
2. Como foi realizado o projeto
3. Explicação do código
   1. Uso do Registro
   2. Função main
   3. Função lerArquivo
   4. Função print
   5. Função insercaoDados
   6. Função atualizacaoDados
   7. Função buscaDados
   8. Função ordenacaoDadosCadastrados
4. Considerações finais
5. **Introdução**

Este trabalho tem como finalidade colocar em prática todo o conhecimento adquirido nas aulas de Introdução aos Algoritmos (GCC-224), tanto nas aulas práticas quanto nas teóricas.

Os requisitos principais para o trabalho era desenvolver um sistema para cadastro, com operações para:

* inserção de dados;
* atualização de dados;
* busca de dados;
* listagem dos dados cadastrados em ordem crescente.

O tema escolhido foi *games*.

1. **Como foi realizado o projeto**

O grupo decidiu utilizar a plataforma G*itHub* para gerenciar o projeto. Com ela, o grupo foi capaz de ter um controle além de trabalhar em equipe vendo cada alteração feita no arquivo do trabalho.

O *link* para o acesso do nosso projeto no *GitHub* é esse [*aqui*](https://github.com/lucas54neves/games).

1. **Explicação do código**
   1. **Uso do Registro**

Como o grupo falaria sobre games, foi montado um registro (*struct*) chamado *game* utilizando suas subcategorias (nome, ano de lançamento, plataforma e descrição) como campos*.*

* 1. **Função main**

Na função principal (*main*) foi colocada a seguinte linha *setlocale(LC\_ALL, “Portuguese”);* para que fosse possível utilizar palavras do idioma português na saída padrão *(cout)* para os menus e qualquer comunicação com o usuário.

A variável *dadosGame* do tipo *game* foi declarada e um ponteiro foi criado, pois um dos requisitos do trabalho era a utilização de variável dinâmica.

Foi criado um laço *(while)* que é o menu principal. Ele funciona enquanto a opção escolhida pelo usuário não for *0* (que é a opção de fechar o programa). Para que o laço fosse inicializado, a variável opção foi iniciada com o valor -1.

O *system(“clear”)* foi utilizado para limpar a tela toda vez que se abre uma opção do menu.

No menu, foi dado ao usuário cinco opções e solicitado a ele que digite um número referente à opção desejada. De acordo com a opção selecionada pelo usuário, um *switch case* foi utilizado para chamar as funções da opção desejada.

* 1. **Função lerArquivo**

Foi decidido utilizar uma função apenas para a leitura dos dados. Após confirmar com a professora Juliana de que era permitido ler o arquivo em uma função, desde que os dados lidos não sejam trabalhados nessa função, o grupo decidiu que assim seria feito para uma melhor legibilidade do código.

Essa função pega o tamanho do arquivo, descobre a quantidade de jogos e lê jogo por jogo, colocando na variável bloco e salva a quantidade de jogos que o arquivo possui por parâmetro por referência.

* 1. **Função print**

Para uma melhor legibilidade do código, também foi decido criar uma função específica para a impressão dos jogos cadastrados. Essa função *print* imprime o vetor *dadosGame[]* com o tamanho exato de jogos cadastrados.

* 1. **Função insercaoDados**

Na função para inserção de dados, as funções da linguagem *cin.ignore()* e *cin.getline(variável, capacidade do vetor char)* foram utilizadas para o problema ler frases ou nomes compostos sem que os espaços atrapalhem a leitura.

* 1. **Função atualizacaoDados**

Para atualizar os dados, uma função que usa um número de identificação para selecionar o jogo desejado para atualizar foi criada. Esse número foi chamado de *ID*. Além disso, um controle de confirmação antes da atualização de fato dos dados também foi criado.

* 1. **Função buscaDados**

Para a busca de um ou mais jogos cadastrados, uma função que busca os jogos através do ano foi utilizada. É solicitado ao usuário o ano e o programa imprime os jogos encontrados ou a mensagem de *“Nenhum dado encontrado.”.*

* 1. **Função ordenacaoDadosCadastrados**

Como o método de ordenação Bubble Sort é o método de melhor compreensão entre os membros do grupo, foi decidido utilizar este método para a ordenação crescente com o ano como referência.

1. **Considerações finais**

Este trabalho consolidou os conhecimentos adquiridos durante as aulas, uma vez que foi possível colocar a matéria em prática, além dos exercícios do [Dredd](http://dredd.dcc.ufla.br/).

Além disso, o trabalho foi essencial para que todos os membros do grupo melhorassem suas habilidades de trabalho em equipe. Durante o processo de criação do programa, alguns erros no código foram cometidos devido à quantidade de pessoas codificando o mesmo código e por erro de lógica. Porém, esses erros foram corrigidos com a ajuda mútua dos membros.

Por fim, o tema do trabalho foi essencial para motivar o grupo para realizar o projeto, uma vez que esse tema era de interesse de cada um dos membros.