**MEC-SETEC**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**

**MINAS GERAIS – CAMPUS BAMBUÍ**

**MARCO AURÉLIO MONTEIRO LIMA**

**ALGORITMOS E ESTRUTURAS DE DADOS II**

**BAMBUÍ, 2016**

**MARCO AURÉLIO MONTEIRO LIMA**

**ALGORITMOS E ESTRUTURAS DE DADOS II**

Árvores binárias, registros e estruturas de dados solicitado pelo professor: Gabriel da Silva.

**BAMBUÍ, 2016**

PROBLEMA ESCOLHIDO

Cadastro de séries. O programa feito, cadastra e salva o progresso de episódios assistidos em séries, novelas animes etc.

Ex:

Código de cadastro da série: 101

Nome da série: The\_Walking\_Dead

Temporadas completas: 5

Assistindo a: 6º temporada, episodio: 10

ABSTRAÇÃO DA SOLUÇÃO

Foi utilizado as estruturas de pilha e árvore binária para manipular os registros cadastrados, no final do programa é possível salvar os dados em bloco de notas. O primeiro bloco de notas tem o nome de “Séries” ele salva as informações que podem ser consultadas posteriormente e outro com o nome “save” que é possível carregar os dados cadastrados após o programa ser reiniciado.

JUSTIFICATIVAS DE USO DE CADA TAD E FUNÇÕES

* struct TSerie: esse TAD contém os campos necessário para o cadastro.
* struct TCelula: esse TAD contém um ponteiro para manipulação de dados em alocação dinâmica.
* struct TPilha: esse TAD possui dois ponteiros para que o registro tenha comportamento de pilha.
* struct TNoh e struct TArvore: TADS necessários para que os registros tenham comportamento de árvore.
* void IniciarArvore: essa função inicia uma árvore vazia para evitar lixos de memória.
* void CriaPilhaVazia: essa função inicia uma pilha vazia para evitar lixos de memória.
* void inserirArvore: essa função insere um registro na árvore.
* bool PilhaEVazia: verifica se a pilha está sem registros.
* bool ArvoreEVazia: verifica se a árvore está sem registros.
* void MostrarTopo: verifica o último registro empilhado.
* void empilharSeries: empilha os registros na pilha.
* TSerie DesempilhaEPega: esse procedimento desempilha o topo da pilha e retorna o registro desempilhado.
* bool VerificaIfEIgual: Verifica se o código de cadastro do registro que está prestes a ser empilhado já existe na pilha pois o campo código é o campo chave do registro.
* void CadastroDeSeries: essa função coleta os dados para ser empilhado.
* void PesquisarSerie: essa função faz uma busca na árvore através do código de cadastro e retorna os dados do registro se ele existir.
* void MostrarPilha: essa função exibi ao usuario os dados empilhados na pilha.
* void Desempilhar: essa função desempilha o registro que está no topo da pilha e o deleta.
* void pegarAnteriorArvore: essa função reorganiza a árvore depois que um registro é retirado.
* void retirar: essa função retira um determinado arquivo da árvore informando o seu código de cadastro.
* void TransferirPilhaParaArvore: essa função desempilha um item e pega e depois, o insere na árvore. A transferência é feita até que não haja mais registros na pilha.
* void PercorrerInOrdem: essa função visita primeiro a sub-árvore esquerda, depois a raiz e por último a sub-árvore direita.
* void PercorrerPreOrdem: essa função visita primeiro a raiz, depois a sub-árvore esquerda e por último a sub-árvore direita.
* void PercorrerPosOrdem: essa função visita primeiro a sub-árvore esquerda, depois a sub-árvore direita e por último a raiz.
* void MostrarArvore: essa função exibi os dados cadastrados nas três sequências acima.
* void CadastrarSeriesArvore: essa função coleta os dados para serem inseridos na árvore.
* void PercorrerAndEmpilhar: essa função percorre a árvore e coleta os dados para serem empilhados sem que árvore perca seus dados.
* void GravarEmTXT: essa função grava os dados em dois blocos de nota um com o nome “Séries” contendo as informações cadastradas de modo que o usuário possa ler e entender posteriormente e o outro salvo com o nome de “save” para que os dados possam ser carregados depois após o programa ser reiniciado sem perder os dados cadastrados.
* void CarregarUltimoCadastro: essa função caso tenha sido salvo algum progresso ela empilha os registros em uma pilha.
* int MenuArvore: diz respeito as funções e procedimentos que serão feitos na árvore.
* int MenuPilha: diz respeito as funções e procedimentos que serão feitos na pilha.