



Princípios de Programação Procedimental

MANUAL DO PROGRAMADOR

Cláudio André Ventura Alves - 2016225581

João Manuel Leitão - 2016227984

ÍNDICE

1) Introdução	2
2) Organização dos Menus.....	3
3) Estrutura/Organização do Código	4
4) Funções	5

INTRODUÇÃO

Este documento tem como objetivo principal clarificar a estrutura do projeto de princípios de programação procedimental, projeto esse desenvolvido com recurso ao compilador CodeBlocks e feito para correr em ambiente Windows.

O Projeto consiste no desenvolvimento de uma aplicação para gerir os exames do DEI, tendo em conta a utilização desta mesma por funcionários responsáveis na inserção de informações relevantes.

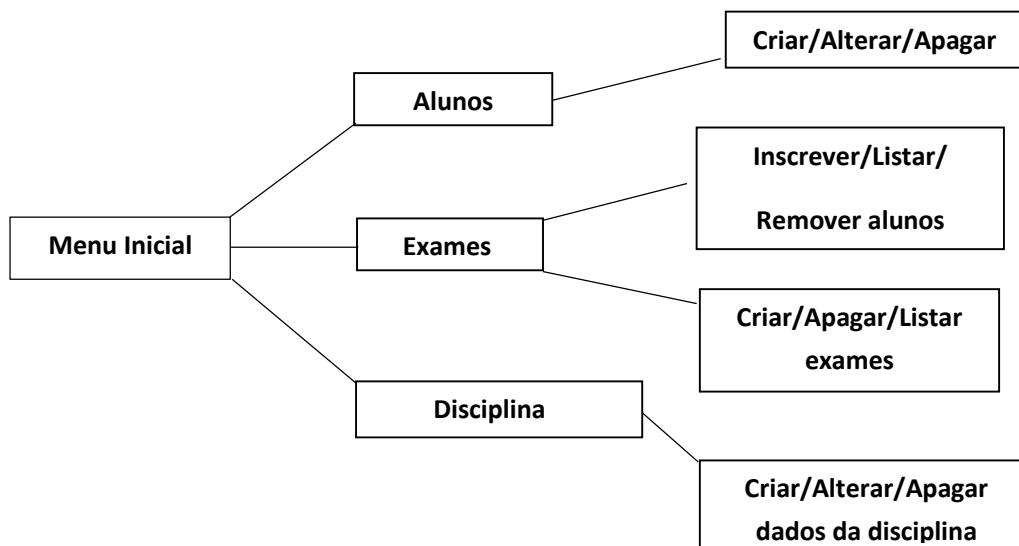
Para tal é necessário criar e armazenar um registo para cada aluno, possibilitando também ao utilizador a sua alteração ou remoção. Em termos de disciplinas temos a possibilidade de criar, sendo esta constituída por nome do docente e nome da mesma, alterar ou apagar. Nos exames, existe a possibilidade de criar, apagar ou listar exames, bem com a possibilidade de inscrever ou remover alunos de exames.

A fim de implementar o pretendido recorreremos a listas ligadas e a ficheiros para armazenar a informação.

ORGANIZAÇÃO DOS MENUS

Escolhemos separar a *User interface* em mais do que um menu de modo a não sobrecarregar o utilizador com todas as opções e funcionalidades existentes, permitindo uma navegação fluida, rápida e lógica através do programa, sendo bastante fácil encontrar o que se procura.

Abaixo encontra-se um esquema dos menus que mostra como estes estão dispostos e como navegá-los.



ESTRUTURA/ORGANIZAÇÃO DO CÓDIGO

Dividimos o programa em vários ficheiros C de forma a facilitar a criação do código do programa, contendo cada um várias funções referentes a cada uma das opções dos diversos menus.

1. **main.c** -> ficheiro onde se encontra o menu inicial,
2. **estruturas.c** -> ficheiro onde se encontram as funções de criação de lista de alunos, de disciplina e de exames.
3. **manipula_listas.c** -> ficheiro onde se encontram as principais funções utilizadas no funcionamento da aplicação.
4. **ficheiros.c** -> ficheiro onde se encontram as funções para carregar os ficheiros de alunos, exames e disciplinas.

Para além dos ficheiros C foi adicionado um ficheiro H (header file) “estruturas.h” onde implementámos o código de todas as estruturas (struct files) existentes no programa de modo a que pudessem estar acessíveis a todas as funções e outros ficheiro H com as inicializações das funções.

FUNÇÕES

MAIN.c

1. Void main ()

Função que contém o menu inicial que suporta todo o programa. O utilizador tem 3 opções: Alunos, onde pode criar, alterar ou apagar dados de alunos; Exames, onde há a possibilidade de criar ou apagar exames, bem inscrever ou remover alunos de exames ou Listar os exames ou alunos; Disciplina, local onde poderá criar, alterar ou apagar dados das disciplinas.

Estruturas.c

1. **Lista_exames * cria_lista_exames():** função utilizada para criar listas de exames.
2. **Lista_alunos * cria_lista_alunos():** função utilizada para criar listas de alunos.
3. **Lista_disciplinas * cria_lista_disciplinas():** função utilizada para criar listas de disciplinas.

Manipula_Listas.c

1. **insere_lista_ordenada(Lista_alunos* node):** função para a inserção de alunos de forma ordenada numa lista vazia.
2. **int lista_vazia_alunos(Lista_alunos *li):** função para limpar uma lista de alunos.
3. **int lista_vazia_disciplinas(Lista_disciplinas *li):** função utilizada para limpar uma lista de disciplina.
4. **int lista_vazia_exames(Lista_exames *li):** função utilizada para limpar uma lista de exame.
5. **int remove_aluno(Lista_alunos *node, int num):** função utilizada para remover um aluno específico, através do seu numero, duma lista.

6. **int insere_disciplina(Lista_disciplinas *node):** função utilizada para a inserção dos dados da disciplina.
7. **int remove_disciplina(Lista_disciplinas *node, char nome):** função utilizada para apagar uma disciplina.
8. **int procura_aluno_dados(Lista_alunos *node, int posicao):** função utilizada para procurar um aluno, pela sua posição na lista.

Ficheiros.c

1. **int carrega_lista_alunos(Lista_alunos *node):** função utilizada para carregar a lista de alunos para o ficheiro.
2. **int carrega_ficheiro_alunos(Lista_alunos *node):** função utilizada para carregar o ficheiro com os alunos.
3. **int carrega_lista_disciplinas(Lista_disciplinas *node):** função utilizada para carregar a lista de disciplinas para o ficheiro.
4. **int carrega_ficheiro_disciplina(Lista_disciplinas *node):** função utilizada para carregar o ficheiro com as disciplinas.