

Registro Cabeçalho (*header record*)

Prof. Me. Sérgio Pelicano Jr

Disciplina: Lab Estrutura de Dados

Departamento de Ciências de Computação e Estatística

E-mail: sergio.pelicano01@etec.sp.gov.br



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"



IBILCE / UNESP - CÂMPUS DE SÃO JOSÉ DO RIO PRETO



- ⦿ Número de registros
- ⦿ Tamanho de cada registro
- ⦿ Nomes dos campos de cada registro
- ⦿ Tamanho dos campos
- ⦿ Datas de criação e atualização



```
#include <stdio.h>
#include <stdlib.h>

typedef struct{
    char tarefa[60];
    int dia, mes, ano;
}ToDo;
```



```
int cadastrar_todo(ToDo **t, int quant, int tam){  
  
    if(quant < tam){  
        ToDo *novo = malloc(sizeof(ToDo));  
  
        printf("\nDigite a Tarefa: ");  
        scanf("%50[^\n]", novo->tarefa);  
    }  
}
```



```
    printf("\nDigite a data prevista dd mm aaaa: ");
    scanf("%d%d%d", &novo->dia, &novo->mes, &novo->ano);
    getchar();
    t[quant] = novo;
    return 1;
}
else{
    printf("\n\tERRO! Vetor cheio!\n");
    return 0;
}
}
```



```
void imprimir(ToDo **t, int quant){
    int i;

    printf("\n\t----- Lista de Tarefas ----- \n");
    for(i = 0; i < quant; i++){
        printf("\t%d = %2d/%2d/%4d\t%s\n", i+1,
t[i]->dia, t[i]->mes, t[i]->ano, t[i]->tarefa);
    }
    printf("\t----- \n");
}
```



```
void alterar_todo(ToDo **t, int quant){  
    int id;  
  
    imprimir(t, quant);  
    printf("\n\tDigite a tarefa a ser alterada: \n");  
    scanf("%d", &id);  
    getchar();  
    id--;
```



```
if(id >= 0 && id < quant){
    ToDo *novo = malloc(sizeof(ToDo));
    printf("\nDigite a nova tarefa: ");
    scanf("%50[^\n]", novo->tarefa);
    printf("\nDigite a data da tarefa dd mm aaaa: ");
}
else
    printf("\n\tCodigo invalido!\n");
}
}
```




```
void salvar(ToDo **t, int quant, char arq[]){  
    FILE *file = fopen(arq, "w");  
    int i;  
  
    if(file){  
        fprintf(file, "%d\n", quant);  
        for(i = 0; i < quant; i++){  
            fputs(t[i]->tarefa, file);  
            fputc('\n', file);  
        }  
    }  
}
```



```
fprintf(file, "%d %d %d\n", t[i]->dia, t[i]  
        ->mes, t[i]->ano);  
}  
fclose(file);  
}  
else  
    printf("\n\tERRO AO ABRIR/CRIAR O  ARQUIVO!\n");  
}
```

Registro Cabeçalho (header record)



```
int ler_arquivo(ToDo **c, char arq[]){
    FILE *file = fopen(arq, "r");
    int quant = 0, i;
    ToDo *novo;
    if(file){
        fscanf(file, "%d\n", &quant);
        for(i = 0; i < quant; i++){
            novo = malloc(sizeof(ToDo));
            fscanf(file, "%50[^\n]", novo->tarefa);
            fscanf(file, "%d %d %d\n", &novo->dia,
                &novo->mes, &novo->ano);
            c[i] = novo;
        }
    }
}
```



```
    }  
    else  
        printf("\n\tERROR AO ABRIR/CRIAR O  
                ARQUIVO!\n");  
    return quant;  
}
```



```
int main(){
    ToDo *tarefa[100];
    char arquivo[] = {"ToDo.txt"};
    int opcao, tam = 100, quant = 0;

    do{
        printf("\n\t0 - Sair\n\t1 - Cadastrar\n\t2 -
                Alterar\n\t3 - Imprimir\n\t4 -
                Salvar\n\t5 - Ler arquivo\n");
        scanf("%d", &opcao);
        getchar();
    }
```



```
switch(opcao){  
    case 1:  
        quant += cadastrar_todo(tarefa, quant, tam);  
        break;  
    case 2:  
        alterar_todo(tarefa, quant);  
        break;  
    case 3:  
        imprimir(tarefa, quant);  
        break;  
}
```



```
case 4:
    salvar(tarefa, quant, arquivo);
    break;
case 5:
    quant = ler_arquivo(tarefa, arquivo);
    break;
default:
    if(opcao != 0)
        printf("\n\tOpcao invalida!!!\n");
}
```



```
    }while(opcao != 0);  
  
    return 0;  
}
```




1) Usando como base o exemplo da aula, incluir um campo para Chave Primaria das tarefas e criar um arquivo de index que agilize a localização do registro correspondente a tarefa.

2) Identifique suas dificuldades com a Linguagem e C e estrutura de dados.

Enviar para: sergio.pelicano01@etec.sp.gov.br até 29/10