

Atividade de ordenação da lista por arquivo

**** Nota:** O arquivo alunos.txt não estava nos anexos da atividade, apenas o dadosEnem.txt, que tem um formato de estrutura diferente. Por isso, eu fiz a lista encadeada baseada no formado desse arquivo.

```
salgado2004@LeoSalgado3420:~/ufpr/edIII$ ./exercioEnem
Lendo dados de escolas...
Total de escolas lidas: 30228
Tempo de leitura: 0.03 segundos
Tempo de ordenação por conteúdo: 1.39 segundos
Tempo de ordenação por posição: 1.31 segundos
Escola 1 (CAMPO LARGO - PR), Tipo State; Médias: {'Ciências exatas': 440.63, 'Matemática': 426.57, 'Ciências humanas': 490.53, 'Linguagens': 459.95, 'Redação': 482.04}
Escola 2 (OURICURI - SP), Tipo Private; Médias: {'Ciências exatas': 572.77, 'Matemática': 558.95, 'Ciências humanas': 504.60, 'Linguagens': 678.69, 'Redação': 669.73}
Escola 3 (MIRANDOPOLIS - SP), Tipo Private; Médias: {'Ciências exatas': 563.55, 'Matemática': 558.83, 'Ciências humanas': 591.44, 'Linguagens': 672.40, 'Redação': 684.00}
Escola 4 (SANTA FE DO SUL - SP), Tipo Private; Médias: {'Ciências exatas': 611.77, 'Matemática': 585.66, 'Ciências humanas': 623.48, 'Linguagens': 706.64, 'Redação': 637.27}
Escola 5 (PRESIDENTE EPITACIO - SP), Tipo State; Médias: {'Ciências exatas': 428.53, 'Matemática': 435.73, 'Ciências humanas': 494.47, 'Linguagens': 485.77, 'Redação': 518.31}
Escola 6 (SAO BERNARDO DO CAMPO - SP), Tipo State; Médias: {'Ciências exatas': 466.92, 'Matemática': 467.62, 'Ciências humanas': 526.31, 'Linguagens': 532.66, 'Redação': 514.44}
Escola 7 (AURIFLAMA - SP), Tipo Private; Médias: {'Ciências exatas': 562.52, 'Matemática': 557.23, 'Ciências humanas': 585.12, 'Linguagens': 627.89, 'Redação': 635.56}
Escola 8 (LINS - SP), Tipo State; Médias: {'Ciências exatas': 440.57, 'Matemática': 446.38, 'Ciências humanas': 501.93, 'Linguagens': 476.67, 'Redação': 534.30}
Escola 9 (FORTALEZA - CE), Tipo Private; Médias: {'Ciências exatas': 506.52, 'Matemática': 506.66, 'Ciências humanas': 551.80, 'Linguagens': 527.57, 'Redação': 585.33}
Escola 10 (SAO PAULO - SP), Tipo Private; Médias: {'Ciências exatas': 500.63, 'Matemática': 500.59, 'Ciências humanas': 558.61, 'Linguagens': 554.53, 'Redação': 578.71}
Escola 11 (GUARATUBA - PR), Tipo State; Médias: {'Ciências exatas': 445.53, 'Matemática': 451.97, 'Ciências humanas': 506.30, 'Linguagens': 490.62, 'Redação': 484.76}
Escola 12 (LIMOEIRO - PE), Tipo Private; Médias: {'Ciências exatas': 599.46, 'Matemática': 589.05, 'Ciências humanas': 606.65, 'Linguagens': 675.83, 'Redação': 610.26}
Escola 13 (FORTALEZA - CE), Tipo Private; Médias: {'Ciências exatas': 601.03, 'Matemática': 581.87, 'Ciências humanas': 599.41, 'Linguagens': 719.85, 'Redação': 667.93}
Escola 14 (FORTALEZA - CE), Tipo Private; Médias: {'Ciências exatas': 602.12, 'Matemática': 590.49, 'Ciências humanas': 603.52, 'Linguagens': 711.87, 'Redação': 683.69}
Escola 15 (ITAPECERICA DA SERRA - SP), Tipo Private; Médias: {'Ciências exatas': 579.19, 'Matemática': 566.77, 'Ciências humanas': 610.70, 'Linguagens': 634.42, 'Redação': 577.65}
```

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

```
typedef struct node{
    int matricula;
    char uf[3];
    char cidade[50];
    char categoria[15];
    float medias[5];
    struct node *prox;
} escola;
```

```
FILE *abreArquivo(char *arquivo, char *modo){
    FILE *fp;
    if (!(fp = fopen(arquivo, modo))){
        printf("Falha ao abrir o arquivo!");
        exit(1);
    }
    return fp;
}
```

```
void insereEscola(escola **lista, escola *escolaAux){
    escola *aux = *lista;
    if(*lista == NULL){
        *lista = escolaAux;
        escolaAux->prox = NULL;
    }else{
        escolaAux->prox = *lista;
        *lista = escolaAux;
    }
}
```

```

    }
}

int leArquivo(escola **lista){
    printf("Lendo dados de escolas...\n");
    FILE *fp = abreArquivo("ordenacaoDadoEnem/DadosEnem.txt", "r");
    escola *escolaAux;
    int count = 0;
    while(!feof(fp)){
        escolaAux = (escola*)malloc(sizeof(escola));
        char linha[256];
        if (fgets(linha, sizeof(linha), fp) != NULL) {
            sscanf(linha, "%d;%[^;];%[^;];%[^;];%f;%f;%f;%f;%f",
                &escolaAux->matricula,
                escolaAux->uf,
                escolaAux->cidade,
                escolaAux->categoria,
                &escolaAux->medias[0],
                &escolaAux->medias[1],
                &escolaAux->medias[2],
                &escolaAux->medias[3],
                &escolaAux->medias[4]);
        }
        insereEscola(lista, escolaAux);
        count++;
    }
    fclose(fp);
    return count;
}

void printEscolas(escola *lista){
    escola *aux = lista;
    for(int i=0; aux != NULL && i < 15; i++){
        printf("Escola %d (%s - %s), Tipo %s; Médias: {'Ciências exatas': %.2f,
'Matemática': %.2f, 'Ciências humanas': %.2f, 'Linguagens': %.2f, 'Redação':
%.2f}\n",
            aux->matricula, aux->cidade, aux->uf, aux->categoria, aux->medias[0], aux-
>medias[1], aux->medias[2], aux->medias[3], aux->medias[4]);
        aux = aux->prox;
    }
}

escola* ordenacaoPorConteudo(escola *lista){
    escola *aux = lista;
    escola *aux2 = lista;
    escola *menor;
    while(aux != NULL){

```

```

    menor = aux;
    aux2 = aux->prox;
    while(aux2 != NULL){
        if(aux2->matricula < menor->matricula){
            menor = aux2;
        }
        aux2 = aux2->prox;
    }
    int matricula = aux->matricula;
    aux->matricula = menor->matricula;
    menor->matricula = matricula;
    char uf[3];
    strcpy(uf, aux->uf);
    strcpy(aux->uf, menor->uf);
    strcpy(menor->uf, uf);
    char cidade[50];
    strcpy(cidade, aux->cidade);
    strcpy(aux->cidade, menor->cidade);
    strcpy(menor->cidade, cidade);
    char categoria[15];
    strcpy(categoria, aux->categoria);
    strcpy(aux->categoria, menor->categoria);
    strcpy(menor->categoria, categoria);
    float medias[5];
    for(int i=0; i<5; i++){
        medias[i] = aux->medias[i];
        aux->medias[i] = menor->medias[i];
        menor->medias[i] = medias[i];
    }
    aux = aux->prox;
}
return lista;
}

```

```

escola* ordenacaoPorPosicao(escola *lista){
    escola *aux = lista;
    escola *aux2 = lista;
    escola *menor;
    while(aux != NULL){
        menor = aux;
        aux2 = aux->prox;
        while(aux2 != NULL){
            if(aux2->matricula < menor->matricula){
                menor = aux2;
            }
            aux2 = aux2->prox;
        }
    }
}

```

```

        escola *ponteiro = aux->prox;
        aux->prox = menor->prox;
        menor->prox = ponteiro;
        aux = aux->prox;
    }
    return lista;
}

double calculaTempo(clock_t t0, clock_t t1){
    return (((double) (t1 - t0)) / CLOCKS_PER_SEC);
}

int main(void){
    escola *listaEscolas;
    clock_t t0, t1;

    t0 = clock();
    int len = leArquivo(&listaEscolas);
    t1 = clock();
    double tempoLeitura = calculaTempo(t0, t1);
    printf("Total de escolas lidas: %d\n", len);
    printf("Tempo de leitura: %.2f segundos\n", tempoLeitura);

    t0 = clock();
    ordenacaoPorConteudo(listaEscolas);
    t1 = clock();
    double tempoOrdenacaoConteudo = calculaTempo(t0, t1);

    t0 = clock();
    listaEscolas = ordenacaoPorPosicao(listaEscolas);
    t1 = clock();
    double tempoOrdenacaoPosicao = calculaTempo(t0, t1);

    printf("Tempo de ordenação por conteúdo: %.2f segundos\n",
tempoOrdenacaoConteudo);
    printf("Tempo de ordenação por posição: %.2f segundos\n",
tempoOrdenacaoPosicao);
    printEscolas(listaEscolas);
}

```