

```
/**
 * Arquivo: meu_editor.c
 * Versão : 1.0
 * Data   : 2024-10-18 23:22
 * -----
 * Este programa implementa um editor simples de buffers, utilizado para testar
 * a abstração bufferTAD. O editor lê e executa comandos simples informados pelo
 * usuário.
 *
 * Baseado em: Programming Abstractions in C, de Eric S. Roberts.
 *             Capítulo 9: Efficiency and ADTs (pg. 379-380).
 *
 * Prof.: Abrantes Araújo Silva Filho (Computação Raiz)
 *        www.computacaoraiz.com.br
 *        www.youtube.com.br/computacaoraiz
 *        github.com/computacaoraiz
 *        twitter.com/ComputacaoRaiz
 *        www.linkedin.com/company/computacaoraiz
 *        www.abrantes.pro.br
 *        github.com/abrantesasf
 */

/**** Includes: ****/

#include "buffer.h"
#include <ctype.h>
#include "genlib.h"
#include <stdio.h>
#include <stdlib.h>
#include "simpio.h"

/**** Declarações de Subprogramas: ****/

static void executar_comando (bufferTAD buffer, string linha);
static void ajuda (void);

/**** Função Main: ****/

int main (void)
{
    bufferTAD buffer = criar_buffer();

    while (TRUE)
    {
        printf("**");
        executar_comando(buffer, GetLine());
        exhibir_buffer(buffer);
    }

    liberar_buffer(&buffer);
}

/**** Definições de Subprogramas: ****/

/**
 * Procedimento: executar_comando
 * Uso: executar_comando(buffer, linha);
 * -----
 * Faz o parser do comando informado pelo usuário, na linha, e executa esse
 * comando no buffer.
 */
```

```
*/

static void executar_comando (bufferTAD buffer, string linha)
{
    switch (toupper(linha[0]))
    {
        case 'I':
            for (int i = 1; linha[i] != '\0'; i++)
                inserir_caractere(buffer, linha[i]);
            break;
        case 'D': apagar_caractere(buffer); break;
        case 'F': mover_cursor_para_frente(buffer); break;
        case 'B': mover_cursor_para_tras(buffer); break;
        case 'J': mover_cursor_para_inicio(buffer); break;
        case 'E': mover_cursor_para_final(buffer); break;
        case 'H': ajuda(); break;
        case 'Q': exit(0); break;
        default: printf("Comando inválido.\n"); break;
    }
}

/**
 * Procedimento: ajuda
 * Uso: ajuda( );
 * -----;
 * Lista os comandos disponíveis no editor.
 */

static void ajuda (void)
{
    printf("Use os seguintes comandos para editar o buffer:\n");
    printf(" I...   Insere o texto informado após a letra \'I\'.\n");
    printf("  F      Move o cursor 1 caractere para frente.\n");
    printf("  B      Move o cursor 1 caractere para trás.\n");
    printf("  J      Move o cursor para o início do buffer.\n");
    printf("  E      Move o cursor para o final do buffer.\n");
    printf("  D      Apaga o próximo caractere.\n");
    printf("  H      Exibe esta ajuda.\n");
    printf("  Q      Sai do programa.\n");
}
```