

```
/**
 * Arquivo: buffer.h
 * Versão : 1.0
 * Data   : 2024-10-18 22:51
 * -----
 * Este arquivo define uma interface para a abstração de um editor de buffer,
 * implementado como um TAD.
 *
 * Baseado em: Programming Abstractions in C, de Eric S. Roberts.
 *             Capítulo 9: Efficiency and ATDs (pg. 373-378).
 *
 * 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
 */

/**** Inicia Boilerplate da Interface ****/

#ifndef _BUFFER_H
#define _BUFFER_H

/**** Includes ****/

#include "genlib.h"

/**** Tipos de Dados ****/

/**
 * TIPO: bufferTAD
 * -----
 * Este tipo abstrato de dado é utilizado para representar um editor de buffer.
 */

typedef struct bufferTCD *bufferTAD;

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

/**
 * FUNÇÃO: criar_buffer
 * Uso: buffer = criar_buffer( );
 * -----
 * Esta função aloca memória de modo dinâmico, em quantidade suficiente para a
 * representação interna do bufferTAD, e inicializa o buffer para representar
 * um buffer vazio.
 */

bufferTAD criar_buffer (void);

/**
 * PROCEDIMENTO: liberar_buffer
 * Uso: liberar_buffer(buffer);
 * -----
 * Este procedimento libera o espaço de armazenamento alocado para o buffer. O
 * argumento deve ser um PONTEIRO para o buffer (um ponteiro para ponteiro para
 * struct bufferTCD).
 */
```

```
*/

void liberar_buffer (bufferTAD *buffer);

/**
 * PROCEDIMENTOS: mover_cursor_para_frente
 *                  mover_cursor_para_tras
 * Uso: mover_cursor_para_frente(buffer);
 *      mover_cursor_para_tras(buffer);
 * -----
 * Estes procedimentos movem o cursor para frente e para trás, no buffer, um
 * caractere por vez. Se "mover_cursor_para_frente" for chamada no final do
 * buffer, ou se "mover_cursor_para_tras" for chamada no início do buffer, os
 * procedimentos não têm efeito nenhum.
 */

void mover_cursor_para_frente (bufferTAD buffer);
void mover_cursor_para_tras (bufferTAD buffer);

/**
 * PROCEDIMENTOS: mover_cursor_para_final
 *                  mover_cursor_para_inicio
 * Uso: mover_cursor_para_final(buffer);
 *      mover_cursor_para_inicio(buffer);
 * -----
 * Estes procedimentos movem o cursor para o final ou para o início do buffer,
 * respectivamente.
 */

void mover_cursor_para_final (bufferTAD buffer);
void mover_cursor_para_inicio (bufferTAD buffer);

/**
 * PROCEDIMENTO: inserir_caractere
 * Uso: inserir_caractere(buffer, c);
 * -----
 * Insere o caractere "c" no buffer "buffer", na posição atual do cursor. Após
 * a inserção o cursor é posicionado após o caractere inserido, para permitir
 * inserções consecutivas.
 */

void inserir_caractere (bufferTAD buffer, char c);

/**
 * PROCEDIMENTO: apagar_caractere
 * Uso: apagar_caractere(buffer);
 * -----
 * Apaga o caractere imediatamente posterior ao cursor. Se o cursor já está no
 * final do buffer, não causa nenhum efeito.
 */

void apagar_caractere (bufferTAD buffer);

/**
 * PROCEDIMENTO: exibir_buffer
 * Uso: exibir_buffer(buffer);
 * -----
 * Exibe o conteúdo atual do buffer no terminal.
 */
```

```
void exhibir_buffer (bufferTAD buffer);  
  
/*** Finaliza Boilerplate da Interface ***/  
  
#endif
```