

```
/**
 * Arquivo: scanner.c
 * Versão : 1.0
 * Data   : 2024-10-17 23:45
 * -----
 * Este arquivo implementa a interface scanner.h.
 *
 * Baseado em: The Art and Science of C, de Eric S. Roberts.
 *             Capítulo 10: Modular Development (pg. ).
 *
 * 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
 */

#include <ctype.h>
#include "genlib.h"
#include "scanner.h"
#include "simpio.h"
#include "strlib.h"

/**** Estado Interno do Scanner: ****/

static string buffer;
static int tamanho_buffer;
static int posicao;

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

/**
 * PROCEDIMENTO: iniciar_scanner
 * Uso: iniciar_scanner(string);
 * -----
 * Este procedimento apenas inicializa as variáveis do estado interno do scanner
 * (o estado encapsulado).
 */

void iniciar_scanner (string linha)
{
    buffer = linha;
    tamanho_buffer = StringLength(linha);
    posicao = 0;
}

/**
 * FUNÇÃO: obter_proximo_token
 * Uso: str = obter_proximo_token( );
 * -----
 * Se o próximo caractere é alfanumérico (letra ou dígito), a função continua
 * lendo para encontrar um "pedaço" contínuo de caracteres e retorna esse token.
 * Se o caractere atual não for alfanumérico (se for um espaço ou pontuação),
 * uma string contendo apenas esse caractere é retornada.
 */

string obter_proximo_token (void)
```

```
{
    char c;
    int inicio;

    if (posicao >= tamanho_buffer)
        Error("Não há mais tokens.");

    c = IthChar(buffer, posicao);
    if (isalnum(c))
    {
        inicio = posicao;
        while (posicao < tamanho_buffer && isalnum(IthChar(buffer, posicao)))
            posicao++;
        return SubString(buffer, inicio, posicao - 1);
    }
    else
    {
        posicao++;
        return CharToString(c);
    }
}

/**
 * PREDICADO: final_da_linha
 * Uso: if(final_da_linha( )) . . .
 * -----
 * Compara se a posição atual do buffer com o comprimento.
 */

bool final_da_linha (void)
{
    return posicao >= tamanho_buffer;
}
```