mar 19, 19 16:28 **paso6_wordscounter.h** Page 1/1

```c
#ifndef __WORDSCOUNTER_H__
#define __WORDSCOUNTER_H__

#include <string.h>
#include <stdio.h>

// Tipo wordscounter_t: procesa cantidad de palabras dentro de un archivo.
typedef struct {
    size_t words;
} wordscounter_t;

// Initializa la instancia self para ser utilizada
// Pre: self apunta un sector vÃ¡lido de memoria
void wordscounter_create(wordscounter_t *self);

// Destruye la instancia self liberando sus recursos.
// Pre: self fue inicializado mediante wordscounter_create
void wordscounter_destroy(wordscounter_t *self);

// Retorna la cantidad de palabras procesadas
// Pre: self fue inicializado mediante wordscounter_create
size_t wordscounter_get_words(wordscounter_t *self);

// Procesa el contenido de text_file, contando sus palabras.
// Pre: self fue inicializado mediante wordscounter_create.
//      text_file es un archivo vÃ¡lido, abierto para lectura.
void wordscounter_process(wordscounter_t *self, FILE *text_file);

#endif

```

mar 19, 19 16:28 **paso6_wordscounter.c** Page 1/1

```c
#include "paso6_wordscounter.h"
#include <string.h>
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>

#define STATE_WAITING_WORD 0
#define STATE_IN_WORD 1
#define STATE_FINISHED 2
#define DELIM_WORDS " ,.;:\n"

// Compara el caracter leÃ-do c y define el nuevo estado.
static char wordscounter_next_state(wordscounter_t *self, char state, char c);

void wordscounter_create(wordscounter_t *self) {
    self→words = 0;
}

void wordscounter_destroy(wordscounter_t *self) {
    //do nothing
}

size_t wordscounter_get_words(wordscounter_t *self) {
    return self→words;
}

void wordscounter_process(wordscounter_t *self, FILE *text_file) {
    char state = STATE_WAITING_WORD;
    do {
        int c = getc(text_file);
        state = wordscounter_next_state(self, state, c);
    } while (state ≠ STATE_FINISHED);
}

static char wordscounter_next_state(wordscounter_t *self, char state, char c) {
    char next_state = state;

    if (state ≡ STATE_WAITING_WORD) {
        if (c ≡ EOF) {
            next_state = STATE_FINISHED;
        } else if (strchr(DELIM_WORDS, c) ≡ NULL) {
            next_state = STATE_IN_WORD;
        }
    } else if (state ≡ STATE_IN_WORD) {
        if (c ≡ EOF) {
            next_state = STATE_FINISHED;
            self→words++;
        } else if (strchr(DELIM_WORDS, c) ≠ NULL) {
            self→words++;
            next_state = STATE_WAITING_WORD;
        }
    }

    return next_state;
}

```

mar 19, 19 16:28 **paso6_main.c** Page 1/1

```c
#include <string.h>
#include <stdio.h>
#include <stdbool.h>
#include "paso6_wordscounter.h"

#define SUCCESS 0
#define ERROR 1

int main(int argc, char* argv[]) {
    FILE* input;
    if (argc > 1) {
        input = fopen(argv[1], "r");
    } else {
        input = stdin;
    }

    if (¬ input) {
        return ERROR;
    } else {
        wordscounter_t counter;
        wordscounter_create(&counter);
        wordscounter_process(&counter, input);
        size_t words = wordscounter_get_words(&counter);
        printf("%zu\n", words);
        wordscounter_destroy(&counter);
        if (input ≠ stdin)
            fclose(input);
        return SUCCESS;
    }
}

```

mar 19, 19 16:28 **Table of Content** Page 1/1