

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include "types.h"
5  #include "config.h"
6  #include "errors.h"
7  #include "process.h"
8  #include "main.h"
9
10
11  /*****Prototipos*****/
12  status_t validate_args(int argc, char *argv[], config_t *);
13  /*****/
14
15  extern config_t config;
16
17  int main(int argc, char *argv[]){
18      status_t st;
19
20      if((st=(validate_args(argc,argv, &config))) != OK){
21          print_error_message(st);
22          return EXIT_FAILURE;
23      }
24      if((st = process_gps_data()) != OK){
25          print_error_message(st);
26          return EXIT_FAILURE;
27      }
28      return EXIT_SUCCESS;
29  }
30
31  /*****
32  Funcion para validar argumentos ingresados por linea
33  de comandos. Valida el numero de argumentos ingresados
34  y que sean correctos, luego guarda lo ingresado en una estructura.
35  *****/
36  status_t validate_args(int argc, char *argv[], config_t *config){
37      if(argv == NULL || config == NULL)
38          return ERROR_NULL_POINTER;
39      if(argc != NUMBER_ARGS)
40          return ERROR_INVALID_NUMBER_ARGS;
41      if(strncmp(argv[FORMAT_FLAG_POSITION], FORMAT_FLAG, FORMAT_FLAG_SPAN))
42          return ERROR_INVALID_ARGS;
43      if(!strncmp(argv[FORMAT_TYPE_POSITION], FORMAT_TYPE_GREGORIAN,
44          FORMAT_TYPE_GREGORIAN_SPAN)){
45          (*config).format = GREGORIAN_FORMAT;
46          return OK;
47      }
48      if(!strncmp(argv[FORMAT_TYPE_POSITION], FORMAT_TYPE_JULIAN,
49          FORMAT_TYPE_JULIAN_SPAN)){
50          (*config).format = JULIAN_FORMAT;
51          return OK;
52      }
53      return ERROR_INVALID_ARGS;
54  }
```