

C

TDA



Conceptos

- Modelo
 - Estructura de datos (struct)
 - Comportamiento (funciones)
- “Lifetime”:
 - “Create”
 - “Destroy”
- ¿Quién debe alojar la estructura?
 - “Create”
 - Usuario del TDA

Ejemplo - StackFile.h:

```
#ifndef __FILESTACK__
#define __FILESTACK__
#define _XOPEN_SOURCE 500
#include <stdlib.h>
#include <stdio.h>

typedef struct {
    FILE* file;
    size_t item_size;
} filestack_t;

/* Crea el filestack usando 'filepath' y usando 'item_size' bytes para cada elemento */
void filestack_create(filestack_t *self, const char* filepath, size_t item_size);
/* Pone un elemento ('value') en el stack */
void filestack_put(filestack_t* self, void* value);
/* Saca el primer elemento del filestack y lo pone en 'buffer' (asume que apunta a un
lugar válido) */
void filestack_pop(filestack_t *self, void* buffer);
/* Destruye el filestack y libera los reursos de memoria */
void filestack_destroy(filestack_t *self);

#endif
```

Ejemplo - StackFile.c:

```
#include "filestack.h"
#include <unistd.h>
#include <sys/types.h>

void filestack_create(filestack_t *self, const char* filepath, size_t item_size) {
    self->file = fopen(filepath, "a+");
    self->item_size = item_size;
}

void filestack_put(filestack_t* self, void* value) {
    fwrite(value, self->item_size, 1, self->file);
}

void filestack_pop(filestack_t *self, void* buffer){
    fseek(self->file, -self->item_size, SEEK_END);
    fread(buffer, self->item_size, 1, self->file);
    fseek(self->file, -self->item_size, SEEK_END);
    ftruncate(fileno(self->file), ftell(self->file));
}

void filestack_destroy(filestack_t *self) {
    fclose(self->file);
}
```

Ejemplo - main.c:

```
#include <stdio.h>
#include "filestack.h"
int main(int argc, char* argv[])
{
    filestack_t f;
    int i;

    filestack_create(&f, "pila.dat", sizeof(int));

    for (int value = 0; value < 10; ++value)
        filestack_put(&f, &value);

    for (i = 0; i < 5; ++i)
    {
        int value = 0;
        filestack_pop(&f, &value);
        printf("Value: %d\n", value);
    }
    filestack_destroy(&f);
    return 0;
}
```