

BIBLIOTECAS

¿Qué son y para que sirven?

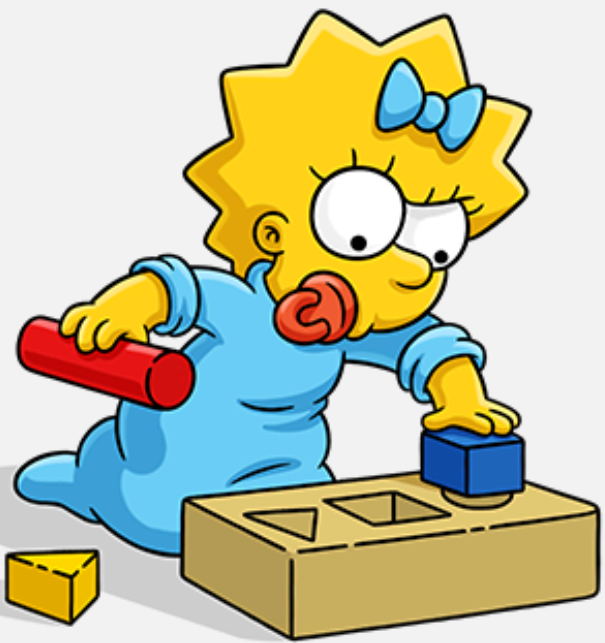


¿QUE VEMOS HOY?

01. ¿Qué es una biblioteca?
02. Relación con lo que venimos viendo
03. ¿Cómo armamos una biblioteca?
04. Funciones privadas y públicas
05. Ejemplo

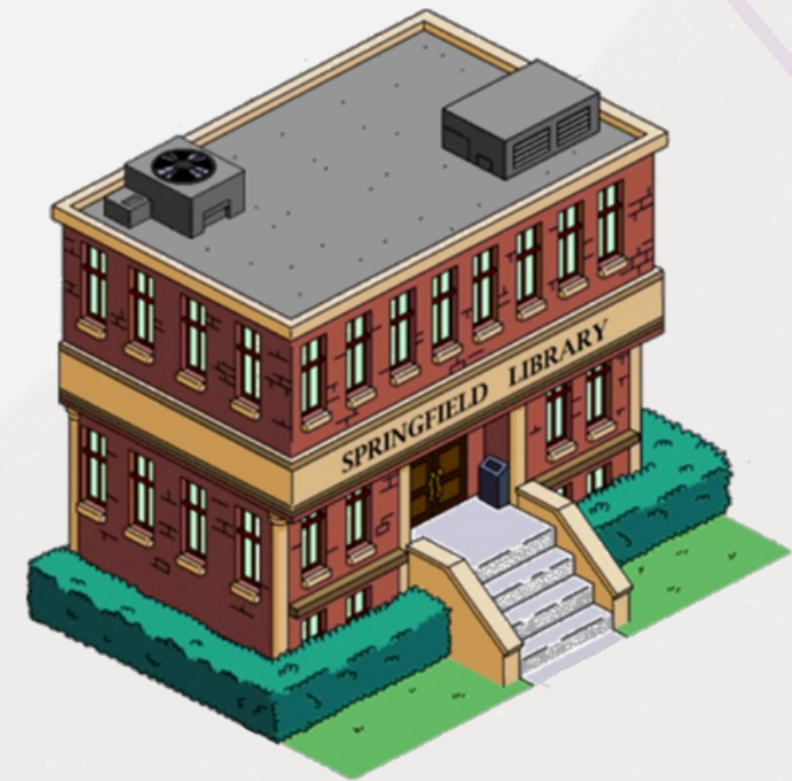
CON LO QUE SABEMOS...

Suponganse que quieren vender su código del
tp0 a alguien
¿Cómo lo harían?



¿QUÉ ES UNA BIBLIOTECA?

Una biblioteca en C es un contrato entre 2 partes
Por un lado una parte proporciona funciones con ciertas condiciones
Y la otra parte utiliza esas funciones sabiendo que hacen pero no
cómo lo hacen



¿LES SUENA A ALGO?



CONDICIONES PARA UNA FUNCIÓN

Ya sabemos hacer que una función tenga condiciones. Las pre y post condiciones nos dicen que necesita nuestra función y que devuelve

SABER EL QUÉ PERO NO EL CÓMO

Este concepto suena bastante parecido a caja negra y caja blanca, donde por un lado sabemos qué con ciertas entradas nos van a devolver ciertas salidas pero no sabemos cómo es que funciona internamente

¿CÓMO SE ARMAN?

Se componen de dos archivos un .h que contiene las firmas de las funciones y un .c que contiene el cuerpo de estas
Normalmente cuando le damos el código a alguien solo puede ver el .h



FUNCIONES PÚBLICAS

Teniendo en cuenta que no siempre van a
ver nuestro .c podemos pensar que las
funciones declaradas en el .h son públicas

FUNCIONES PRIVADAS

Dentro de nuestro .c podemos tener
funciones que sirvan para la
implementación de las funciones de la
biblioteca

VEAMOS UN EJEMPLO





IMPLEMENTACIÓN SOLO CON FUNCIONES PÚBLICAS

```
#ifndef MI_BIBLIO_H
#define MI_BIBLIO_H

#include <stdbool.h>

/*
 * Pre: -
 * Post: Devuelve la suma de num1 y num2
 */
int suma(int num1, int num2);

/*
 * Pre: -
 * Post: Devuelve la resta de num1 y num2 en ese orden
 */
int resta(int num1, int num2);

/*
 * Pre: Num2 no puede ser 0
 * Post: Devuelve la división truncada de num1 y num2
 */
int division(int num1, int num2);

#endif // MI_BIBLIO_H
```

Header Guard

`#ifndef #define #endif`

Sirven para no tener multiples
declaraciones de lo que está dentro de la
biblioteca

```
#include "mi_biblio.h"  
#include <stdio.h>
```

```
int suma(int num1, int num2){  
    return num1 + num2;  
}
```

```
int resta(int num1, int num2){  
    return num1 - num2;  
}
```

```
int division(int num1, int num2){  
    return num1 / num2;  
}
```



IMPLEMENTACIÓN CON FUNCIONES PÚBLICAS Y PRIVADAS

```
#ifndef MI_BIBLIO_H
#define MI_BIBLIO_H

#include <stdbool.h>

/*
 * Pre: -
 * Post: Devuelve la suma de num1 y num2
 */
int suma(int num1, int num2);

/*
 * Pre: -
 * Post: Devuelve la resta de num1 y num2 en ese orden
 */
int resta(int num1, int num2);

/*
 * Pre: No se aceptan divisiones con numeros negativos
 * Post: Devuelve la división truncada de num1 y num2 o -1
        en caso de error
 */
int division(int num1, int num2);

#endif // MI_BIBLIO_H
```

```
#include "mi_biblio.h"
#include <stdio.h>

const int ERROR = -1;

int suma(int num1, int num2){
    return num1 + num2;
}

int resta(int num1, int num2){
    return num1 - num2;
}

bool es_divisor_0(int num2){
    if(num2 == 0){
        return false;
    }
    return true;
}

int division(int num1, int num2){
    if(!es_divisor_0(num2)){
        return num1 / num2;
    }
    printf("Error: No podes dividir por 0 capo.\n");
    return ERROR;
}
```

En este último caso podemos ver que la función de tipo bool “es_divisor_0” no está declarada en el .h pero se utiliza en el .c, por lo tanto es una función privada

COMO COMPILAR

```
$ gcc cuentas.c mi_biblio.c mi_biblio.h -o ejemplo_clase  
$ ./ejemplo_clase
```

Acá cuentas.c sería el archivo con main
que consume la biblioteca



¿PREGUNTAS?

FIN.

