

# Bibliotecas de usuario

martes, 16 de noviembre de 2021 07:12 a. m.

Para crear nuevas directivas es necesario, al menos, generar dos archivos:

1. El primero de ellos llevará la extensión ".h" y se ubicará en el mismo directorio del código principal (el que tiene a la función main).

En este archivo se deben especificar las siguientes secciones de código:

- a. Incluir librerías ANSI (Estándar) como: `stdio.h`, `stdlib.h`, `string.h`
  - b. Macros y constantes que se declaren con los operadores `#` y `##`
  - c. Declaración de estructuras, TDA y prototipo de funciones.
2. Un archivo librería con extensión ".c", el cual tendrá como cabecera al archivo ".h" creado anteriormente. Este archivo tiene por objetivo implementar las funciones declaradas con un prototipo en dicho archivo ".h".
  3. Cuando estén generados ambos archivos, la librería ya estará lista para utilizarse en cualquier programa principal que la requiera. Ahora se debe generar un programa principal, el cual contendrá al `main()`. Los 3 archivos deben de encontrarse en el mismo directorio para poderse compilar y generar el código ejecutable.

Ejemplo:

"Archivo.h"

```
/*Importación de librerías estándar, macros,
declaración de estructuras, tda, variables globales y prototipo de funciones.
*/
#include <stdio.h>
#define CONSTANTE 100
typedef struct{
    int dato;
    //...
} Elemento;
void imprimeElemento(Elemento);
void imprimeElemento2(Elemento*);
```

"Archivo.c"

```
//En este archivo .c se implementan unicamente las funciones descritas como
protipo
//en el archivo .h
#include "miLibreria.h"

/*
    @param var Se necesita una copia de los datos
*/
void imprimeElemento(Elemento var){
    printf("Este es el dato contenido en la estructura: %d\n", var.dato);
}
/*
    @param ptrDato Se necesita la dirección de memoria de la variable a enviar
*/
void imprimeElemento2(Elemento* ptrDato){
    printf("Este es el dato contenido en la estructura: %d\n", ptrDato->dato);
}
```

```
}
```

"ArchivoPrueba.c"

```
#include "miLibreria.h"
int main(int argc, char** argv){
    Elemento info;
    info.dato=44;
    imprimeElemento2(&info);
    imprimeElemento(info);
    return 0;
}
```

Compilación: gcc miLibreria.h miLibreria.c test.c -O3 -o programa

### Segundo Ejemplo:

"Archivo.h"

```
/*Importación de librerías estándar, macros,
declaración de estructuras, tda, variables globales y prototipo de funciones.
*/
#include <stdio.h>
#define TAM 5
typedef struct{
    int alto;
    int ancho;
} Hoja;
typedef struct{
    Hoja hojas[TAM];
} Cuaderno;
void crearTDACuaderno(Cuaderno* miCuaderno, Hoja* arregloHojas);
void imprimeCuaderno(Cuaderno miCuaderno);
void actualizartDatosCuaderno(Cuaderno* miCuaderno, Hoja hoja, int posicion);
```

"Archivo.c"

```
//En este archivo .c se implementan unicamente las funciones descritas como
protipo
//en el archivo .h
#include "miLibreria.h"
/*
    @param miCuaderno Dirección de memoria de la variable cuaderno
    @param arregloHojas Los datos que se almacenaran en el TDA
*/
void crearTDACuaderno(Cuaderno* miCuaderno, Hoja* arregloHojas){
    int indice;
    for(indice=0; indice<TAM; indice++)
        miCuaderno->hojas[indice]=arregloHojas[indice];
}
/*
    @param miCuaderno Se debe enviar una copia de los datos a través de una
```

```

variable (paso por valor)
*/
void imprimeCuaderno(Cuaderno miCuaderno){
    int indice;
    for(indice=0; indice<TAM; indice++){
        printf("El contenido de la hoja # %d es ancho=%d y alto=%d.
\n",indice,miCuaderno.hojas[indice].ancho,miCuaderno.hojas[indice].alto);
    }
}
/*
    @param miCuaderno Dirección de memoria de la variable cuaderno
    @param hoja Hoja que se va a guardar en el arrglo interno de cuaderno
    @param posicion Posicion donde se requiere guardar la hoja enviada
*/
void actualizartDatosCuaderno(Cuaderno* miCuaderno, Hoja hoja, int posicion){
    miCuaderno->hojas[posicion]=hoja;
}

```

```

"ArchivoTest.c"
#include "miLibreria.h"
int main(int argc, char** argv){
    Hoja hoja1={30,40};
    Hoja hoja2={45,40};
    Hoja hoja3={25,25};
    Hoja hoja4={10,12};
    Hoja hojas[TAM];
    hojas[0]=hoja1;
    hojas[1]=hoja2;
    hojas[2]=hoja3;
    hojas[3]=hoja4;
    Cuaderno cuaderno;
    crearTDACuaderno(&cuaderno,hojas);
    imprimeCuaderno(cuaderno);

    Hoja hoja5={4,4};
    actualizartDatosCuaderno(&cuaderno,hoja5,4);
    imprimeCuaderno(cuaderno);
    return 0;
}

```

Compilación: gcc miLibreria.h miLibreria.c test.c -O3 -o programa