



GUIA LABORATORIO DE SISTEMAS DISTRIBUIDOS USO DE PUNTEROS Y ESTRUCTURAS

A continuación se muestra el código y pantallazos de una pequeña aplicación que calcula el promedio de páginas de los capítulos de un libro. Observe los diferentes usos de punteros para apuntar una estructura, variables y el llamado a una función.

```
#include <stdio.h>
#include <stdlib.h>

//structs que contienen los atributos de Libro y Capitulo
struct Capitulo{
    int numeroPaginas;
}typedef Capitulo;

struct Libro{
    char nombreLibro[30];
    int numCapitulosV1;
    Capitulo* capitulosV1;
    float promedioPaginas;
}typedef Libro;

//Funcion para calcular el promedio de paginas de los capitulos
//Ingresa el vector que contiene los capitulos y n: indica el tamaño del vector
float calcularPromedioPaginas(Capitulo* capitulosV1, int n)
{
    int i=0;
    float numeroPaginas=0;
    float promedio=0;
    for(i=0;i<n;i++)
    {
        numeroPaginas=numeroPaginas + capitulosV1[i].numeroPaginas;
    }
    promedio=numeroPaginas/n;
    return promedio;
}

//Declaracion del puntero que permitira llamar la funcion calcularPromedioPaginas
float (*funcion) (Capitulo* capitulosV1, int n);

main(){

    int opcion = 4;
    //Declaracion de un puntero de tipo Libro
    Libro *libro;

    //banderas que permiten comprobar si se registro un libro y si se ha calculado el promedio de paginas
    int b1 = 0;
    int b2 = 0;
    int i;

    do{
        printf("***** MENU ***** \n");
        printf("| 1. Registrar Libro | \n");
        printf("| 2. Calcular Promedio de paginas de los capitulos | \n");
        printf("| 3. Imprimir Libro | \n");
        printf("| 4. Salir | \n");
        printf("***** \n");
        printf("Seleccione la opcion: ");
        scanf("%d",&opcion);

        switch(opcion){

            //Opcion para registrar libro
            case 1:
                //Separacion de memoria dinamica para el puntero libro
                libro = (Libro*)malloc(sizeof(Libro));
                fflush( stdin);
```



```
printf("Ingrese el Nombre del libro: ");
scanf("%*c%[\n]", libro->nombreLibro);
fflush( stdin );
printf("Ingrese el numero de capitulos para la primera version del libro: ");
scanf("%d",&libro->numCapitulosV1);
fflush( stdin );
printf("\n Registro del numero de paginas por capitulo  \n\n");
libro->capitulosV1 = (Capitulo*)malloc(libro->numCapitulosV1*sizeof(Capitulo));
for(i=0; i<libro->numCapitulosV1; i++){
    printf("Ingrese el numero de paginas del capitulo %d:",i+1);
    scanf("%d", &(libro->capitulosV1[i].numeroPaginas));
}

printf("\n");
b1 = 1;
b2 = 0;

break;

//Opcion para calcular el promedio
case 2:
if(b1 == 1){
    funcion = calcularPromedioPaginas;
    libro->promedioPaginas = funcion(libro->capitulosV1,libro->numCapitulosV1);
    printf("El promedio de paginas por capitulo es: %.2f\n\n",libro->promedioPaginas);
    b2 = 1;
}
else{
    printf("Debe registrar primero el libro: Opcion 1.\n\n");
}
break;

//Opcion que muestra los datos del libro
case 3:
if(b2 == 1){
    printf("Nombre del libro: %s \n",libro->nombreLibro);
    printf("Numero de paginas por capitulo para la version 1 ");
    int i;
    for(i=0; i<libro->numCapitulosV1; i++){
        printf("\nNumero de paginas del capitulo %d: ", i+1);
        printf("%d ",libro->capitulosV1[i].numeroPaginas);
    }
    printf("\nPromedio de paginas por capitulo: %.2f\n\n",libro->promedioPaginas);
}
else{
    printf("\nDebe calcular primero el promedio de paginas de los capitulos: Opcion 2.\n\n");
}
break;

//Opcion para salir de la aplicacion
case 4:
free(libro);
break;
default: break;
}
}while(opcion != 4);
return(0);
}
```



Pantallazos del funcionamiento de la Aplicación:

Menú principal

```
***** MENU *****
| 1. Registrar Libro |
| 2. Calcular Promedio de paginas de los capitulos |
| 3. Imprimir Libro |
| 4. Salir |
*****
```

Registrar libro:

```
Ingrese el Nombre del libro: ingenieria de la sociedad
Ingrese el numero de capitulos para la primera version del libro: 4

Registro del numero de paginas por capitulo

Ingrese el numero de paginas del capitulo 1:7
Ingrese el numero de paginas del capitulo 2:12
Ingrese el numero de paginas del capitulo 3:17
Ingrese el numero de paginas del capitulo 4:14
```

Calculo del promedio:

```
***** MENU *****
| 1. Registrar Libro |
| 2. Calcular Promedio de paginas de los capitulos |
| 3. Imprimir Libro |
| 4. Salir |
*****
Seleccione la opcion: 2
El promedio de paginas por capitulo es: 12.50
```

Mostrando en consola los datos del libro:

```
***** MENU *****
| 1. Registrar Libro |
| 2. Calcular Promedio de paginas de los capitulos |
| 3. Imprimir Libro |
| 4. Salir |
*****
Seleccione la opcion: 3
Nombre del libro: ingenieria de la sociedad

Numero de paginas por capitulo para la version 1
Numero de paginas del capitulo 1: 7
Numero de paginas del capitulo 2: 12
Numero de paginas del capitulo 3: 17
Numero de paginas del capitulo 4: 14
Promedio de paginas por capitulo: 12.50
```



Compilación del programa

Compilar nuestro programa con el siguiente comando:

```
gcc main.c -o ejecutable.out
```

Ejecutamos **ejecutable** con el siguiente comando:

```
./ejecutable.out
```

PRACTICA A REALIZAR:

Partiendo del ejercicio anterior, se debe modificar las opciones:

1. Además del Archivo main.c, cree los archivos funciones.c y funciones.h.
2. En el archivo funciones.h coloque la declaración de la función **calcularPromedioPaginas** y los struct **Capitulo** y **Libro**. En el archivo funciones.c ubique la implementación de la función **calcularPromedioPaginas**.
3. Registrar no solo el nombre del libro sino también su código.
4. Cree una struct versión, la cual contenga los siguientes atributos:
int numCapitulos;
Capitulo* capitulos;
5. Cree un vector de versiones en la estructura libro. El vector debe tener 3 elementos.
6. El promedio de páginas del libro debe calcularse mediante la suma del promedio de las páginas de cada versión y su división por tres.

Por ejemplo:

Se tienen 3 versiones del libro, en cada una de las versiones se tienen 2 capítulos:

Versión 1 del libro:

Capitulo1:

No de páginas: 9

Capitulo2:

No de páginas: 8

Promedio de páginas de la versión: 8.5

Versión 2 del libro:

Capitulo1:

No de páginas: 7

Capitulo2:

No de páginas: 6

Promedio de páginas de la versión: 6.5

Versión 3 del libro:

Capitulo1:

No de páginas: 7



Capitulo2:

No de páginas: 10

Promedio de páginas de la versión: 8.5

Promedio de páginas del libro: $(8.5+6.5+8.5) / 3 = 7.83$

7. La información del libro que debe imprimirse es su nombre, código, y por cada versión del libro imprimir para cada capítulo el número de páginas y el promedio de páginas por versión. Al final imprimir el promedio de páginas del libro. (Siga el ejemplo mostrado en el punto 4)
8. Para la compilación de la practica cree un archivo makefile como el visto en la práctica anterior.
9. Comprimir el directorio que contenga todos los archivos de la práctica en un archivo con formato de nombre: `lsd_rpc_r3_apellidoN.rar` y subirlo a la plataforma.

Nota: No puede utilizar variables globales. Debe separar la vista (`main.c`) de las funciones del negocio.

Comprimir el directorio utilizando el comando:

`tar cfvz directorioEmpaquetado.tar.gz directorioParaComprimir`