



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* TISTA GARCÍA EDGAR

*Asignatura:* ESTRUCTURA DE DATOS Y ALGORITMOS I

*Grupo:* 1

*No de Práctica(s):* 3] TIPO DE DATO ABSTRACTO

*Integrante(s):* MURRIETA VILLEGAS ALFONSO

*Semestre:* 2018 - 2

*Fecha de entrega:* 05 DE MARZO DE 2018

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

ALUMNO: ALFONSO MURRIETA

1

# APLICACIONES DE APUNTADORES

## 1] Objetivo de la práctica

- Utilizarás estructuras en lenguaje C para modelar tipos de dato abstracto e implementarlos en las estructuras de datos lineales.

## 2] Desarrollo

### INTRODUCCIÓN

Anteriormente hemos utilizado tipos de datos ya definidos dentro del correspondiente lenguaje de programación, sin embargo, los programadores no solamente hacen uso de este tipo de datos puesto sobretodo limita como abordar los problemas que se pueden presentar ante ellos, para ello existen los **TDA** o tipo de dato abstracto el cual es un conjunto de datos u objetos que son creados de manera personalizada, además de que este tipo de dato se puede manipular de la misma manera que se hace con los tipos de datos normales\*

Como punto relevante hay que mencionar que en el particular caso de C la manera en que podemos utilizar este tipo de dato es mediante el uso de estructuras *struct*.

Para declarar una estructura la manera en la que se hace en C es mediante la palabra *struct* seguida del nombre de la estructura y posteriormente entre llaves las variables ya sean números o cadenas

```
struct nodo {  
    tipoDato elemento1;  
    tipoDato elemento2;  
    ...  
    tipoDato elementoN;  
};
```

### 2.1] Análisis de los problemas (Propuestos)

#### EJERCICIO I

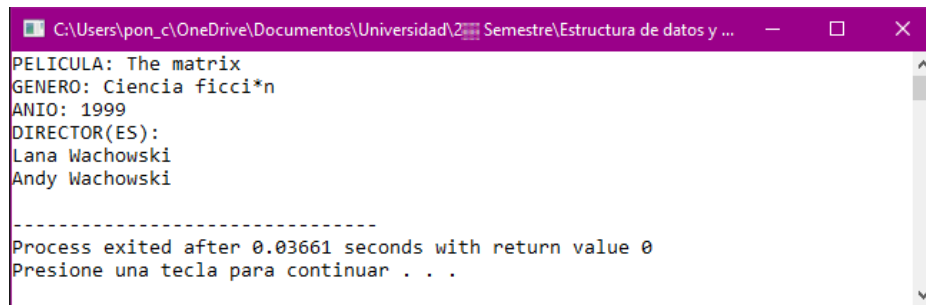
El presente programa es un claro ejemplo de cómo se deben usar los tipos de datos abstractos, sobretodo porque mediante este se basa todo el programa, sin embargo, este programa presenta un problema al compilar y es que mencionar que no reconoce la estructura, para ello lo primero que se debe ver es como está compuesto el programa.

Lo primero que podemos notar es que la función principal *int main* hace uso de la estructura película además de la función ***imprimirDatosPelícula***, también podemos notar que las estructuras utilizadas son del tipo película puesto así se definieron desde el principio del código.

Sin embargo, al indagar en el apartado de la estructura *llenarDatosPelicula* podemos darnos cuenta de que la estructura trabaja sobre estructuras del tipo nodo, para ello lo único que debemos hacer es cambiar este apartado por *struct pelicula* pues es el tipo de estructuras que se están utilizando

```
27 struct pelicula llenarDatosPelicula(char *nombre, char *genero, short anio, short numDirectores)
28
29 struct pelicula movie;
30 movie.nombre = nombre;
31 movie.genero = genero;
32 movie.anio = anio;
33 movie.numDirectores = numDirectores;
34
35 int cont;
36 for (cont = 0 ; cont < movie.numDirectores ; cont++){
37     movie.directores[cont] = directores[cont];
38 }
39
40 return movie;
41 }
```

Una vez que cambiado este pequeño detalle y también de la declaración de esta estructura al inicio del código, podemos darnos cuenta de que el resultado final es sin duda el siguiente:



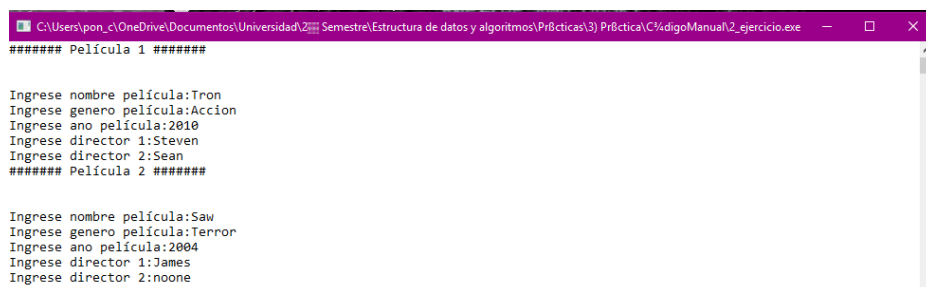
```
C:\Users\pon_c\OneDrive\Documentos\Universidad\2º Semestre\Estructura de datos y ...
PELICULA: The matrix
GENERO: Ciencia ficci*n
ANIO: 1999
DIRECTOR(ES):
Lana Wachowski
Andy Wachowski
-----
Process exited after 0.03661 seconds with return value 0
Presione una tecla para continuar . . .
```

Recordemos que lo único que hace la función *imprimirDatosPelicula* es solamente imprimir datos.

## EJERCICIO II

Este programa lo que tiene en particular es que a diferencia del anterior no tiene los valores inicializados además de que no solamente se usa una vez la estructura declarada, sino que está definida para 2 veces su uso, otro aspecto relevante es que hace uso de dos funciones las cuales se encargan de primero pedir los datos y guardarlos y segundo de imprimir los datos.

A continuación, ingreso algunos datos (Entrada del programa)



```
C:\Users\pon_c\OneDrive\Documentos\Universidad\2º Semestre\Estructura de datos y algoritmos\Prácticas\3 Práctica\CódigoManual\2_ejercicio.exe
##### Película 1 #####
Ingrese nombre película:Tron
Ingrese genero película:Accion
Ingrese ano película:2010
Ingrese director 1:Steven
Ingrese director 2:Sean
##### Película 2 #####
Ingrese nombre película:Saw
Ingrese genero película:Terror
Ingrese ano película:2004
Ingrese director 1:James
Ingrese director 2:noone
```

Con los datos anteriormente ingresados tenemos como salida del programa la siguiente:

```
C:\Users\pon_c\OneDrive\Documentos\Universidad\2º Semestre\Estructura de datos y algoritmos\Prácticas\3) Práctica\CódigoManual\2_gercicio.exe

##### Contenido del arreglo #####

##### PELICULA 2 #####
PELICULA: Saw
GENERO: Terror
ANO: 2004
DIRECTOR(ES):
James
noone

##### PELICULA 1 #####
PELICULA: Tron
GENERO: Accion
ANO: 2010
DIRECTOR(ES):
Steven
Sean

-----
Process exited after 82.95 seconds with return value 0
Presione una tecla para continuar . . .
```

## 2.2] Análisis de los problemas (Práctica)

### EJERCICIO I

El presente ejercicio sirve tiene como principal funcionalidad ayudar a que el administrador de un comerciante pueda realizar su trabajo de manera más sencilla y ordenada para ello los principales elementos que se necesitan considerar son los siguientes:



Como podemos observar lo primero que debemos tomar en cuenta para nuestro programa es realizar una estructura destinada para cada sección de datos, también y como recomendación de la práctica y del profesor debido a la complejidad y sobre todo a la gran cantidad de datos que se pedirán y requerirán se hizo uso de varias funciones las cuales están destinadas para trabajar con cada tipo de dato abstracto.

En el programa realizado se hicieron uso de las siguientes funciones:

```
void imprimirTienda(tienda tienda1)

tienda creartienda()

seccionT crearseccion()

producto crearproducto()

int main
```

Por ejemplo, en el caso de las funciones `crear` se encargan únicamente de pedir y guardar los datos del tipo de dato abstracto con el que trabajan, como se puede ver en las siguientes imágenes, donde la primera

es el código encargado de recopilar los datos de los productos y la segunda es la muestra de esta parte del código ya en ejecución:

```
31 producto crearproducto(){
32     producto producto1;
33
34     printf("\n\tClave del producto: " );
35     scanf("%d",&producto1.clave);
36     printf("\tNombre del producto: " );
37     scanf("%s",producto1.nombre);
38     printf("\tPrecio del producto: " );
39     scanf("%f",&producto1.precio);
40
41     return producto1; //Regresa una estructura del tipo producto
42 }
```

```
_____ [PRODUCTOS] _____
[PRODUCTO 1]
    Clave del producto: 1
    Nombre del producto: Harina
    Precio del producto: 14
[PRODUCTO 2]
    Clave del producto: 2
    Nombre del producto: Sal
    Precio del producto: 8
[PRODUCTO 3]
    Clave del producto: 3
    Nombre del producto: Azucar
    Precio del producto: 10
```

Por otro lado, la función imprimir Tienda simplemente se encargará de imprimir toda la información guardada por las funciones crear, por último, cabe destacar que en este particular caso la función main solamente sirve para desplegar un pequeño menú para el usuario.

```
C:\Users\pon_c\OneDrive\Documentos\Universidad\2º Semestre\Estructura de d...
===== [CREADOR DE DATOS] =====
Elige una opción:
1) Crear tienda
2) Salir
```

Por último, lo que tenemos como resultado final de todo nuestro programa es lo siguiente:

## OBTENIENDO DATOS:

```
C:\Users\pon_c\OneDrive\Documentos\Universidad\2º ...
_____ [TIENDA] _____
NOMBRE: WALMART
ZONA: SUR
_____ [SECCION 1] _____
    Clave de la categoría: 321
    Nombre de la categoría: ABARROTES
_____ [PRODUCTOS] _____
[PRODUCTO 1]
    Clave del producto: 1
    Nombre del producto: HARINA
    Precio del producto: 12
[PRODUCTO 2]
    Clave del producto: 2
    Nombre del producto: SAL
    Precio del producto: 8
[PRODUCTO 3]
    Clave del producto: 3
    Nombre del producto: AZUCAR
    Precio del producto: 7
```

```
C:\Users\pon_c\OneDrive\Documentos\Universidad\2º ...
_____ [SECCION 2] _____
    Clave de la categoría: 654
    Nombre de la categoría: LAPTOPS
_____ [PRODUCTOS] _____
[PRODUCTO 1]
    Clave del producto: 1
    Nombre del producto: HP
    Precio del producto: 10000
[PRODUCTO 2]
    Clave del producto: 2
    Nombre del producto: ASUS
    Precio del producto: 9540
[PRODUCTO 3]
    Clave del producto: 3
    Nombre del producto: HP
    Precio del producto: 10345.65
_____ [SECCION 3] _____
    Clave de la categoría: 987
    Nombre de la categoría: FRUTAS
_____ [PRODUCTOS] _____
```

```
C:\Users\pon_c\OneDrive\Documentos\Universidad\2...
[SECCION 3]
Clave de la categoría: 987
Nombre de la categoría:FRUTAS
[PRODUCTOS]
[PRODUCTO 1]
Clave del producto: 1
Nombre del producto: PLATANO
Precio del producto: 10
[PRODUCTO 2]
Clave del producto: 2
Nombre del producto: MANZANA
Precio del producto: 15
[PRODUCTO 3]
Clave del producto: 3
Nombre del producto: PERA
Precio del producto: 17
```

Como podemos ver al crear una tienda se nos pide la clave, el nombre y el precio de 3 productos por cada sección, donde cada tienda tiene 3 secciones con su respectiva categoría y clave, además de que para crear una tienda se necesita de un nombre y una zona.

### IMPRIMIENDO DATOS(salida):

Una vez que ya se han llenado todos los datos el programa hace uso de la función `imprimirTienda` la cual es la encargada de recopilar todos los datos anteriormente capturados, a continuación, se muestra la salida final del programa:

```
Seleccionar C:\Users\pon_c\OneDrive\Documentos\Univ...
[DATOS CREADOS]
NOMBRE: WALMART
ZONA: SUR
[SECCION 1]
Clave: 321
CATEGORIA: ABARROTES
[PRODUCTO 1]
Clave del producto 1: 1
Nombre del producto 1: HARINA
Precio del producto 1: 12.0
[PRODUCTO 2]
Clave del producto 2: 2
Nombre del producto 2: SAL
Precio del producto 2: 8.0
[PRODUCTO 3]
Clave del producto 3: 3
Nombre del producto 3: AZUCAR
Precio del producto 3: 7.0
[SECCION 2]
Clave: 654
CATEGORIA: LAPTOPS
[PRODUCTO 1]
Clave del producto 1: 1
Nombre del producto 1: HP
Precio del producto 1: 10000.0
[PRODUCTO 2]
Clave del producto 2: 2
Nombre del producto 2: ASUS
Precio del producto 2: 9540.0
[PRODUCTO 3]
Clave del producto 3: 3
Nombre del producto 3: HP
Precio del producto 3: 10345.7
[SECCION 3]
Clave: 987
CATEGORIA: FRUTAS
[PRODUCTO 1]
Clave del producto 1: 1
Nombre del producto 1: PLATANO
```

NOTA: Lo que se ve en la parte de abajo una vez que se termina la impresión de la tienda es nuevamente el menú principal del programa.

```
[PRODUCTO 1]
    Clave del producto 1: 1
    Nombre del producto 1: PLATANO
    Precio del producto 1: 10.0

[PRODUCTO 2]
    Clave del producto 2: 2
    Nombre del producto 2: MANZANA
    Precio del producto 2: 15.0

[PRODUCTO 3]
    Clave del producto 3: 3
    Nombre del producto 3: PERA
    Precio del producto 3: 17.0

===== [CREADOR DE DATOS] =====

Elige una opción:
1) Crear tienda
2) Salir
```

## 2.3] Relación entre los ejercicios y el tema

Tanto los programas del manual, como los que se realizaron en clase el día viernes como los ejercicios propuestos por el profesor tienen como principal propósito el familiarizarnos con los datos de tipo abstractos y es que este tipo de datos son la base para estructuras más complejas y grandes como pueden ser las pilas y colas.

Por otro lado, también esta práctica me permitió ver otros usos tanto para los arreglos como estructuras que anteriormente vimos en las prácticas pasadas.

## 3] Conclusiones

La presente práctica es un claro ejemplo de como en el mundo de la programación el límite que nos ponemos solo está en nuestra capacidad de imaginación e invención, pues el saber que los tipos de datos abstractos son recursos sumamente utilizados por programadores nos hace pensar que sin duda alguna este tipo de datos se puede emplear para casi cualquier problema.

Algo que se debe destacar es que los ejercicios del manual como los ejercicios realizados por el profesor en clase nos hicieron abordar problemas de una manera muy distinta a la que usualmente estábamos acostumbrados a realizar.

También y como aclaración al momento de hacer la práctica en el laboratorio cometí algunos errores sumamente graves y es que los problemas estaban en como usaba de manera incorrecta tanto las funciones como las estructuras, en el caso de las estructuras la manera en que pude evitar problemas fue utilizando *typedef* para de esta forma evitar volver a mencionar struct al inicio de cada uso de una estructura, por otro lado, con respecto a las funciones me pude percatar que en varias no tomé en cuenta que debían retornar valores debido a que estaba usando paso por valor y no paso por referencia.

Sin duda una práctica que pese solo ha tenido un problema me ha dejado en claro que el mundo de la programación tiene una infinidad de formas en las que se puede abordar y resolver un problema.

## 4] Referencias

- 1) Estructuras de datos en C. Luis Jayanes A, Matilde Fernández A., primera edición, ESPAÑA, McGrawHill 2005.
- 2) Curso de Programación C7 C++. Javier Ceballos, tercera edición. USA, Ra – Ma. Educación 2007
- 3) El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.