



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Edgar Tista García

Asignatura: Estructura de Datos y Algoritmos I

Grupo: 1

No de Práctica(s): 5

Integrante(s): Díaz Hernández Marcos Bryan

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 9

Semestre: 2020-2

Fecha de entrega: 14 de marzo de 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo de la práctica

Revisar las definiciones, características, procedimientos y ejemplos de las estructuras lineales Pila y Cola, con la finalidad de que comprendas sus estructuras y puedas implementarlas.

Introducción

El contenido de la práctica esta diseñado para poder comprender el uso de las pilas y colas, por medio de diagramas con los cuales, me guie para hacer los ejercicios relacionados con la práctica correspondiente.

Ejercicios de la guía de laboratorio

La guía en esta ocasión no contenía ejemplos de código.

Ejercicios del Laboratorio

- Ejercicio 1

El primer ejercicio de la práctica solicitaba el modificar la biblioteca Pila.h, donde se tenían que modificar las funciones básicas con las cuales trabajar una pila, estas eran: push y el pop, además de modificar la biblioteca Cola. h y modificar las funciones desencolar y encolar.

Al modificar las bibliotecas y querer compilarlas, me saltaba el siguiente recuadro en la pantalla, no había ningún problema con el código, solamente se refería a que la biblioteca no se podía correr por ser una biblioteca y que tenia que ser utilizada en un archivo aparte (Imagen 1).

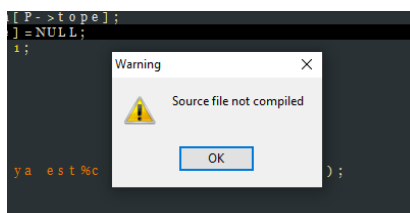
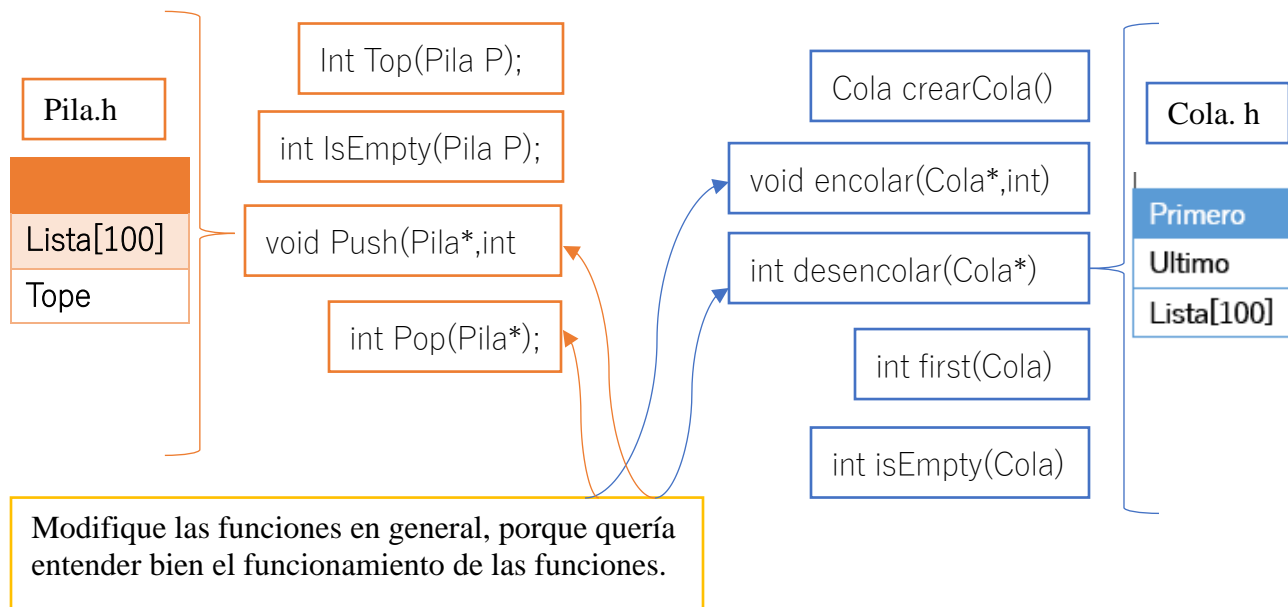


Imagen 1.

Igual al buscar información al respecto esta indicaba que se refería a un error con la versión de compilador, aunque en este caso no afectaba en el código principal.

- Diagrama de funcionamiento



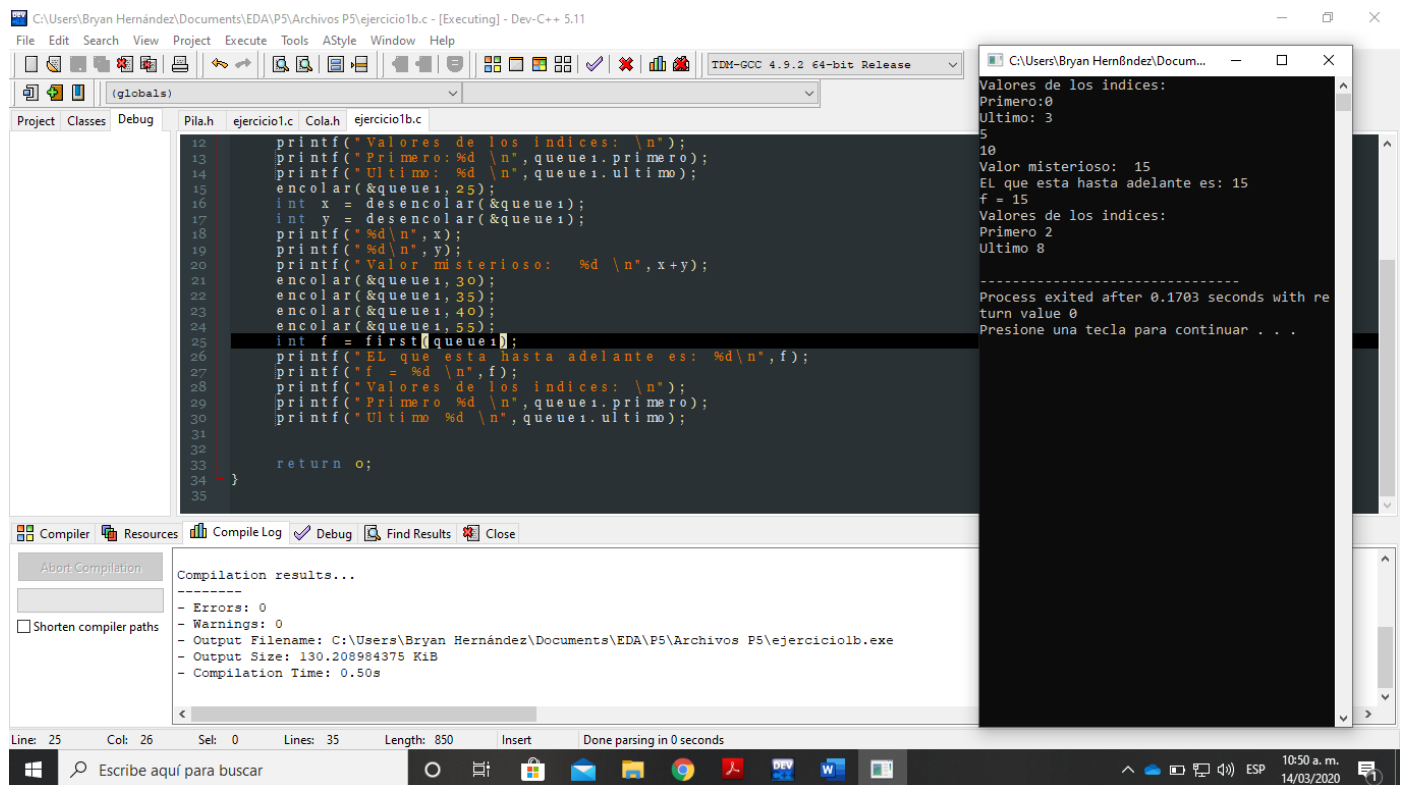
- Relación con teoría.

Para resolver el ejercicio bastaba con tener los apuntes de la clase, por ello no hubo mucha dificultad, únicamente saber el principio de funcionamiento de cada tipo de estructura, y además saber lo que hace cada función. Por ello la relación con la teoría era de un cien por ciento.

Inciso a)

La imagen muestra una captura de pantalla de un IDE de desarrollo de código. El editor principal muestra el código fuente de un programa en C++ que implementa una Pila y una Cola. El código incluye las definiciones de las funciones **CrearPila**, **Push**, **Pop**, **Top**, **IsEmpty**, **crearCola**, **encolar**, **desencolar**, **first** y **isEmpty**. El programa también incluye instrucciones adicionales para probar las funciones. La ventana de la salida muestra el resultado de la ejecución del programa, que incluye el mensaje "vamos a crear una pila", "vamos a ingresar algunos elementos", "El tope de la pila es: 4", "El valor almacenado hasta arriba es: 50", "Ahora el tope de la pila es: 3", "Ahora el valor hasta arriba es: 20", "valor de c es: 30", "La pila ya está vacía.", "Ahora el valor hasta arriba es: -1", "No existen elementos en la pila", "valor de f es: -1", "Process exited after 0.1628 seconds with return value 19", "Presione una tecla para continuar . . .".

Inciso b)



• Ejercicio 2

El segundo ejercicio requería hacer un ordenador de elementos, únicamente mostrara el valor mas grande de 10 elementos, elementos que el usuario insertaba, además de tener que utilizar tres pilas para resolver el ejercicio, con la condición de no utilizar variables externar a las funciones para guardar valores de la pila principal o de cualquiera de las tres pilas.

• Dificultades en el código

Lo más difícil para mi fue el tener que crear el método más adecuado para resolver el problema, debido a que, si se utilizaban las tres pilas y, más de dos contenían más de dos elementos se volvía bastante complejo el buscar una forma de resolver sin utilizar variables extras, en la imagen está parte del código que fue dando orden a los elementos del arreglo (Imagen 1),

```
}
else{
    ((A->lista[A->tope])<(A->lista[A->tope]-1));
    Push(B, Pop(A));
    Push(P, Pop(A));
    Orden(P, A);
}
```

Imagen 1.

Además, al momento de hacer el código me perdí bastante, hasta que llego un momento en que decidí plasmar mis ideas en papel y dar un orden para poder resolver el problema.

- ¿Cómo lo resolví?

En un cuaderno hice varios dibujos de la pila con 10 elementos, y con las funciones de la pila, recordé que se podían pasar valores de una pila a otra pila por medio de pop y del push, con ese principio fue que me guie para poder mover los elementos de una pila a otra. Además, gracias a que se limitó a 10 elementos el orden, fue más fácil poder darle un límite al algoritmo (Imagen 2).

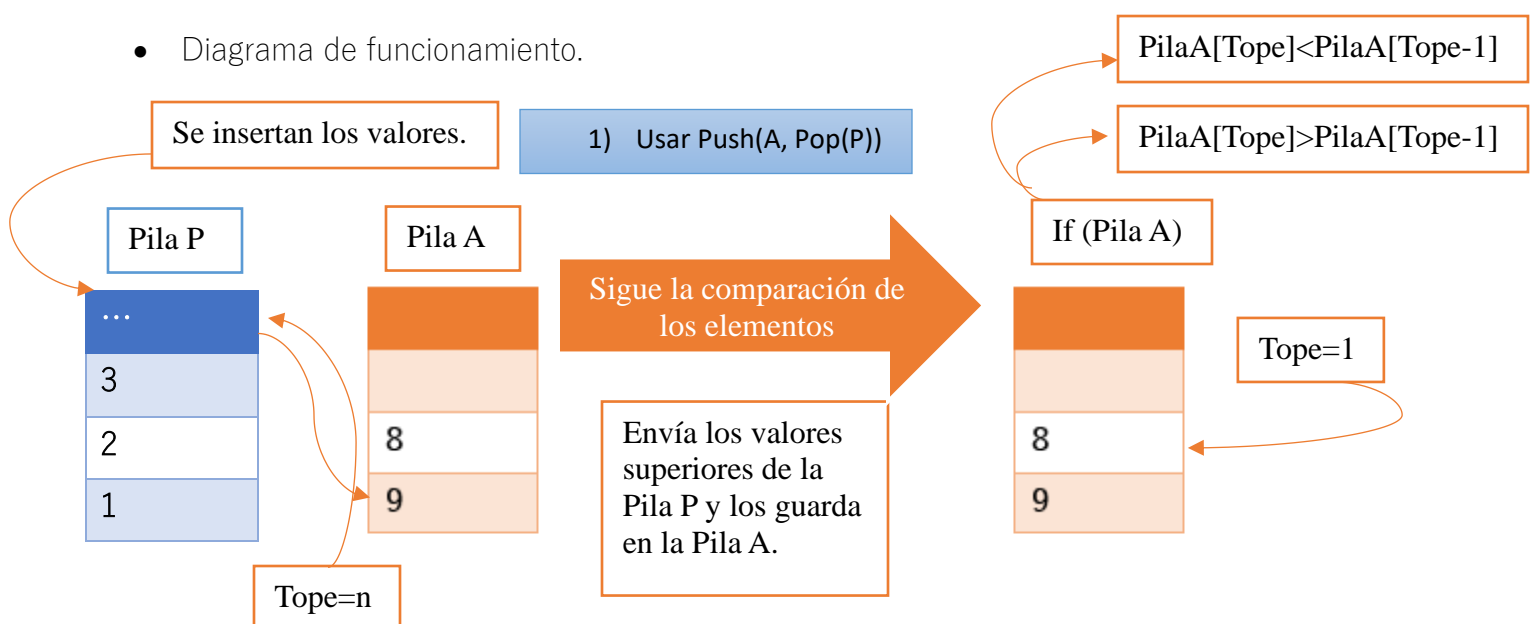
```
void Orden(Pila *P, Pila *A) {  
    int i;  
    for(i=0; i<2; i++){  
        Push(A, Pop(P));  
    }  
}
```

Imagen 2.

Con la función de la imagen anterior podía pasar dos elementos de la Pila P, hacia la Pila A, en la que, hacia una comparación de los elementos, para poder saber cuál era mayor. Una vez comprobado el valor mayor, lo que hacia era mandar el valor mayor de nuevo a P, y el valor menor a la Pila B (Pila de valores pequeños), de esta forma iba separando los valores menores y, a los mayores los iba conservando para poder volver a moverlos de nuevo hacia A y volver a hacer la comparación (Imagen 1).

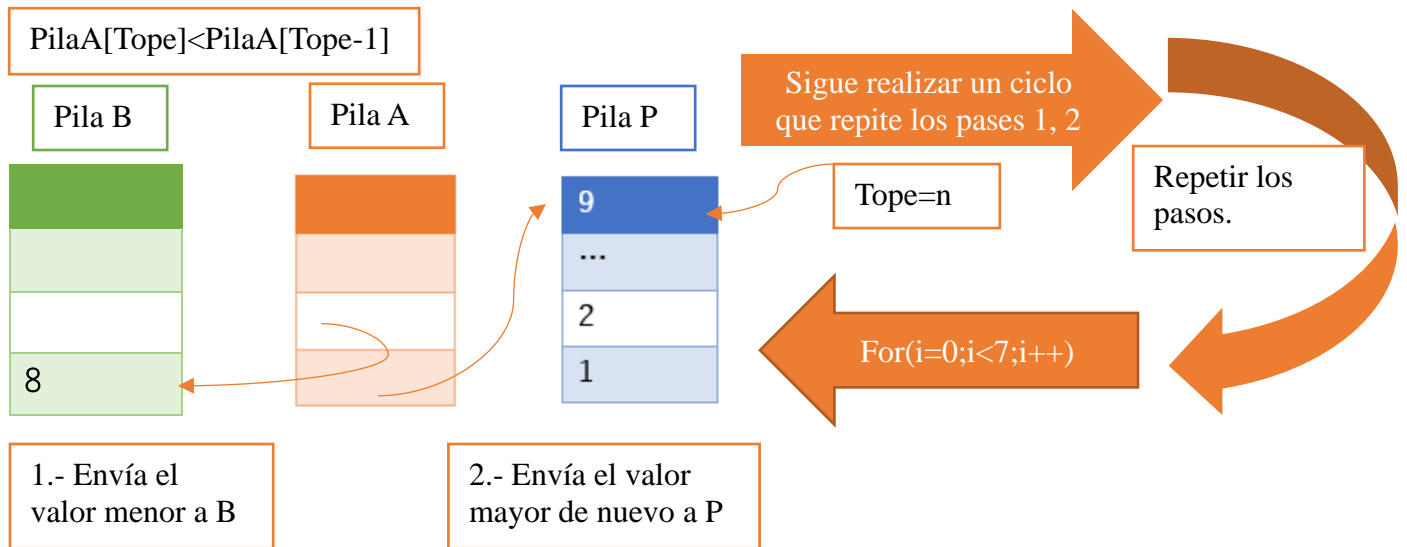
Lo que sucedía al final de la función era que en la Pila P se tenía un único valor que correspondía al valor mayor y en la Pila B se encontraban todos los valores menores, y la Pila A se encontraba vacía ya que esta siempre regresaba los elementos a la pila P, o a la Pila B.

- Diagrama de funcionamiento.

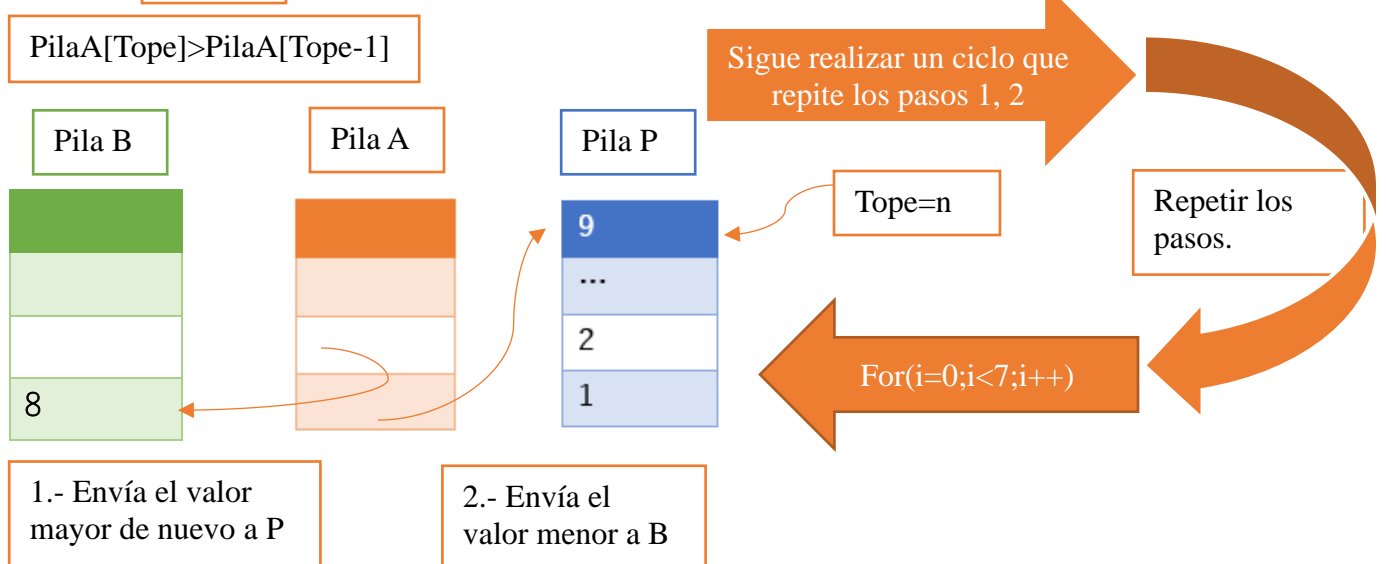


2) Usar Enviar valores

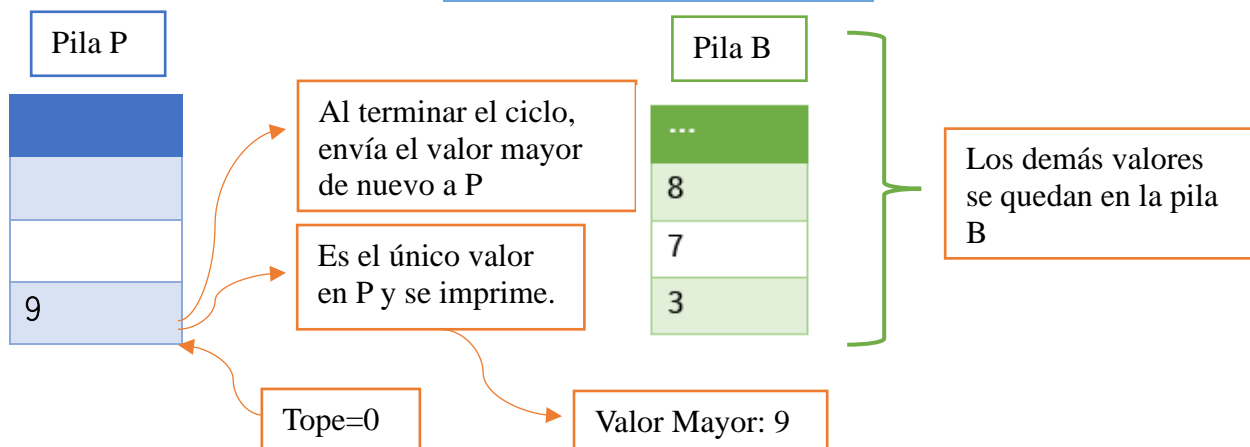
Caso 1.



Caso 2.



3) Imprimir el valor mayor.



- Relación con teoría.

La teoría que envuelve al problema son las funciones de las pilas, porque al saber utilizarlas se pueden recolocar valores de la forma que mas nos convenga, como en este caso, además de utilizar las funciones que, con anterioridad se usaban en estructuras. Lo mas importante es el uso del Pop y el Push ya que gracias a las funciones no son necesarias variables externas para guardar valores. De igual forma en este ejercicio utilice la biblioteca Pila.h para utilizar las funciones declaradas y poder hacer el código más eficiente.

```

C:\Users\Bryan Hernández\Documents\EDA\PS V2\Ejercicio2.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ejercicio1.c Pila.h Ejercicio2.c Cola.h ejercicio1b.c
45     Push(B, Pop(A));
46     Orden(P, A);
47
48
49     else{
50         ((A->lista[A->tope])<(A->lista[A->tope]-
51         Push(B, Pop(A));
52         Push(P, Pop(A));
53         Orden(P, A);
54     }
55
56
57     if((A->lista[A->tope])>(A->lista[A->tope-1])){
58         Push(B, Pop(A));
59         Push(B, Pop(A));
60     }
61
62     else{
63         ((A->lista[A->tope])<(A->lista[A->tope]-
64         Push(B, Pop(A));
65         Push(P, Pop(A));
66     }
67
68     printf("Maximo Valor: %d", Top(*P));
69     public int __cdecl printf(const char* __restrict__, ...)
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Bryan Hernández\Documents\EDA\PS V2\Ejercicio2.c
- Output Size: 130.4208984375 KiB
- Compilation Time: 0.67s
Line: 67 Col: 26 Sel: 0 Lines: 68 Length: 1289 Insert Done parsing in 0.015 seconds
C:\Users\Bryan Hernández\Documents\EDA\PS V2\Ejercicio2.c - [Executing] - Dev-C++ 5.11
Ingresar valores numericos enteros
El valor[1]:
3
El valor[2]:
7
El valor[3]:
9
El valor[4]:
0
El valor[5]:
8
El valor[6]:
15
El valor[7]:
23
El valor[8]:
4
El valor[9]:
5
El valor[10]:
6
Maximo Valor:23
-----
Process exited after 10.17 seconds with return value 0
Presione una tecla para continuar . . .

```

• Ejercicio 3

Los requisitos de ejercicio eran, el pedir al usuario 10 valores que se guardaran en una pila, y por medio de una cola, acomodar los valores de forma inversa a como se insertaron, todas las operaciones sin utilizar una variable extra, únicamente usar las funciones definidas para las colas y pilas, por eso reutilice las bibliotecas definidas en los ejercicios anteriores, para poder realizar le ejercicios más rápido.

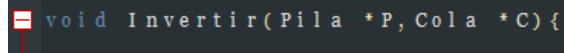
Una de las partes que me hubiera gustado cambiar dentro del código fue la parte donde se solicitaba al usuario cada uno de los valores, donde tuve que hacerlo con código puro, sin poder utilizar una variable externa o un ciclo.

- Dificultades en el código

La parte donde más revolví fue la parte donde tenía que enviar los valores de la pila a la cola, y de la cola a la pila, porque estaba realizando el envío de los valores por medio de los apuntadores, pero los valores no se modificaban, porque solo se estaban copiando los elementos de la pila en la cola.

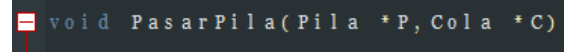
Entonces me puse a reflexionar el como poder enviar un valor de la pila a la cola y que este elemento se pase y vacié el lugar en la lista.

Las partes más difíciles fueron las dos funciones:



```
void Invertir(Pila *P, Cola *C){
```

Imagen 1.

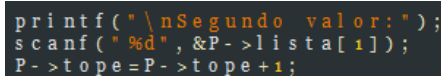


```
void PasarPila(Pila *P, Cola *C){
```

Imagen 2.

- ¿Cómo lo resolví?

Para guardar los valores, utilice los apuntadores y con cada valor iba guardando en su respectivo lugar, además de incrementar el tope de la pila (Figura 3).

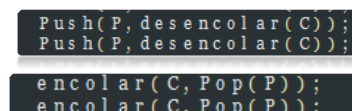


```
printf( "\nSegundo valor:");  
scanf( "%d", &P->lista[1]);  
P->tope=P->tope+1;
```

Figura 3.

El error que cometía estaba relacionado con las funciones porque no las utilizaba simplemente me puse a hacer el código con los contadores, de cada estructura, pero no funcionaba de esa forma, por lo que, decidí tomar un descanso y decidí utilizar la función "encolar" y la función "Push".

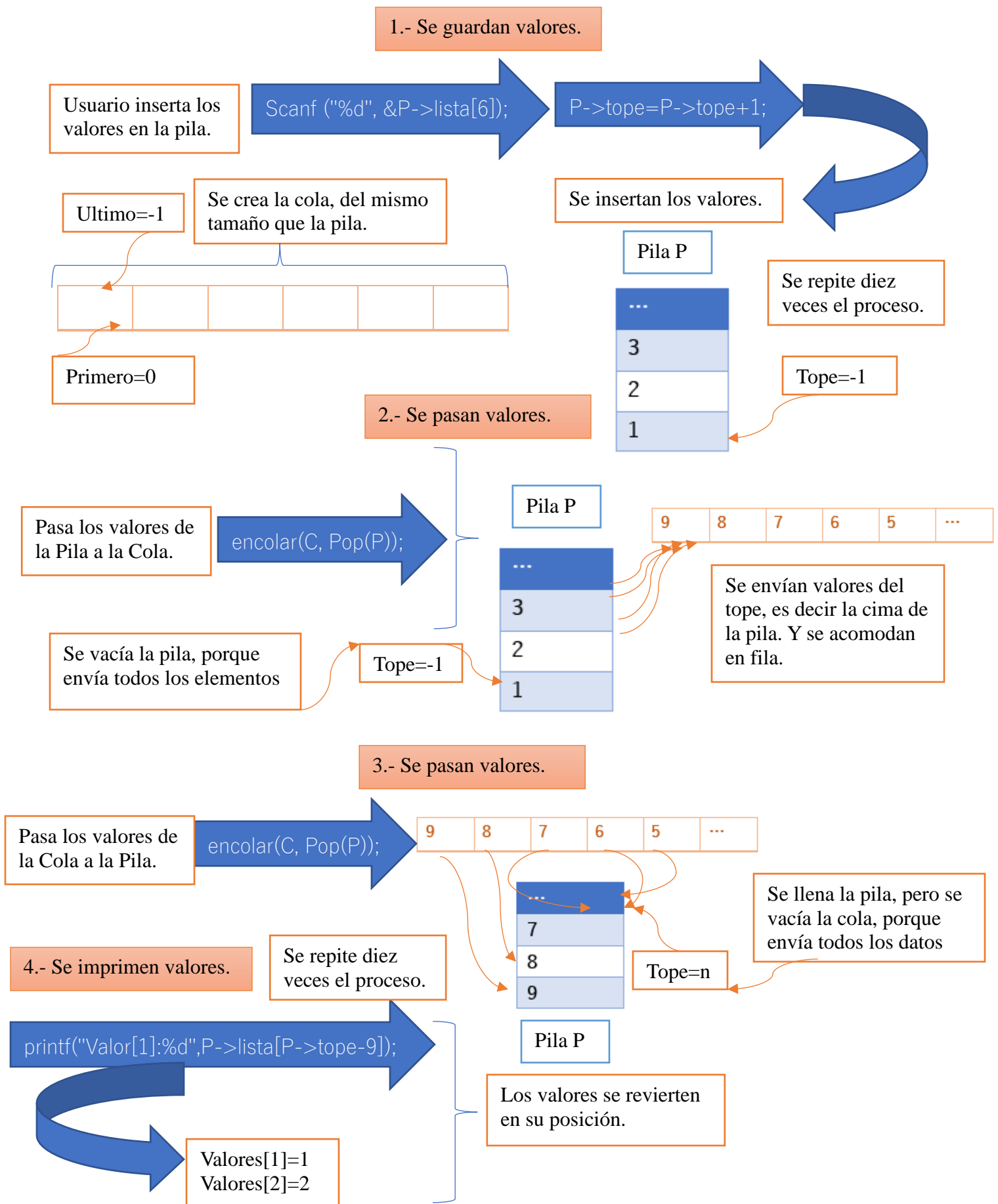
Primero use el encolar para mandar los valores de la Pila a través del pop, y que se almacenaran en la cola, y para después enviar los valores de la cola por medio del desencolar para mandar el argumento al Push, así el orden se invertiría (Figura 4).



```
Push(P, desencolar(C));  
Push(P, desencolar(C));  
  
encolar(C, Pop(P));  
encolar(C, Pop(P));
```

Figura 4.

- Diagrama de funcionamiento



- Relación con teoría.

La teoría aplicada en el ejercicio se relaciona con las funciones de las pilas y las colas, porque con estas se pueden acomodar los valores, de forma rápida, sin necesidad de usar alguna variable, por ello es importante saber que es lo que hace cada función y saber cuándo utilizarla. En este ejercicio me ayudó mucho el utilizar funciones, por ello pude concluir el ejercicio.

```

85 Push(P, desencolar(C));
86 Push(P, desencolar(C));
87 Push(P, desencolar(C));
88 Push(P, desencolar(C));
89 Push(P, desencolar(C));
90 Push(P, desencolar(C));
91 Push(P, desencolar(C));
92
93
94 }
95
96 void Imprimir(Pila *P){
97
98     printf("\nValor[1]: %d\n", P->lista[P->tope-9]);
99     printf("Valor[2]: %d\n", P->lista[P->tope-8]);
100    printf("Valor[3]: %d\n", P->lista[P->tope-7]);
101    printf("Valor[4]: %d\n", P->lista[P->tope-6]);
102    printf("Valor[5]: %d\n", P->lista[P->tope-5]);
103    printf("Valor[6]: %d\n", P->lista[P->tope-4]);
104    printf("Valor[7]: %d\n", P->lista[P->tope-3]);
105    printf("Valor[8]: %d\n", P->lista[P->tope-2]);
106    printf("Valor[9]: %d\n", P->lista[P->tope-1]);
107    printf("Valor[10]: %d\n", P->lista[P->tope]);
108
109 }

```

```

Ingresar
Primer valor:1
Segundo valor:2
Tercer valor:3
Cuarto valor:4
Quinto valor:5
Sexto valor:6
Septimo valor:7
Octavo valor:8
Noveno valor:9
Decimo valor:0
Cola vacia
Valor[1]:0
Valor[2]:9
Valor[3]:8
Valor[4]:7
Valor[5]:6
Valor[6]:5
Valor[7]:4
Valor[8]:3
Valor[9]:2
Valor[10]:1

-----
Process exited after 6.648 seconds with return value 0
Presione una tecla para continuar . . .

```

Compilation results...

```

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Bryan Hernández\Documents\EDA\P5\Díaz Hernández Marcos G1 P5 V2\Ejercicio3.exe
- Output Size: 132.056640625 KiB
- Compilation Time: 0.58s

```

• Ejercicio 4

El último ejercicio consistía en crear un tipo de dato Libro, el cual contendría determinados miembros, además de que cada libro que se insertara en la estructura se debería guardar en la cola de Libros, y para saber el numero de libros a guardar, el usuario debe indicar la cantidad, ello implica memoria dinámica. Al momento de que se leyera los libros, se debe mostrar el tiempo aproximado de lectura, es decir que cada hoja tiene una duración de 20 segundos.

Además, se debe modificar la biblioteca Cola.h, pero en este caso, hice una versión secundaria de Cola.h, a la que llame Cola2.h, porque los programas anteriores contienen a Cola.h como base, entonces si la modifico, crearía problemas en los demás programas.

- Dificultades en el código

La dificultad que tuve fue solamente, el como mostrar el tiempo de lectura de cada libro, pero esto se resolvía con operaciones sencillas de multiplicación, y por ello no tuve mucha complicación en este ejercicio, a comparación de la que tuve en los anteriores.

- ¿Cómo lo resolví?

Para modificar la biblioteca, cambie los tipos de dato de las funciones declaradas en Cola2.h, por el tipo de dato libro, para ello, definí Libro antes de la estructura Cola, para que, en la parte de Lista de la Cola, el tipo de dato fuese Libro, y lo detectara (Imagen 1).

```
typedef struct{
    char Titulo[15];
    char Autor[10];
    float Paginas;
    int Clave;
}Libro;
```

Imagen 1.

Para las instrucciones de insertar libro, llevé a cabo una función donde se pidieran los valores, y esto lo hice dentro de un ciclo for, para hacer más rápido el proceso (Imagen 2).

```
void Llenar(Libro *A, int z, Cola *C){
    int i;
    for(i=0; i<z; i++){
```

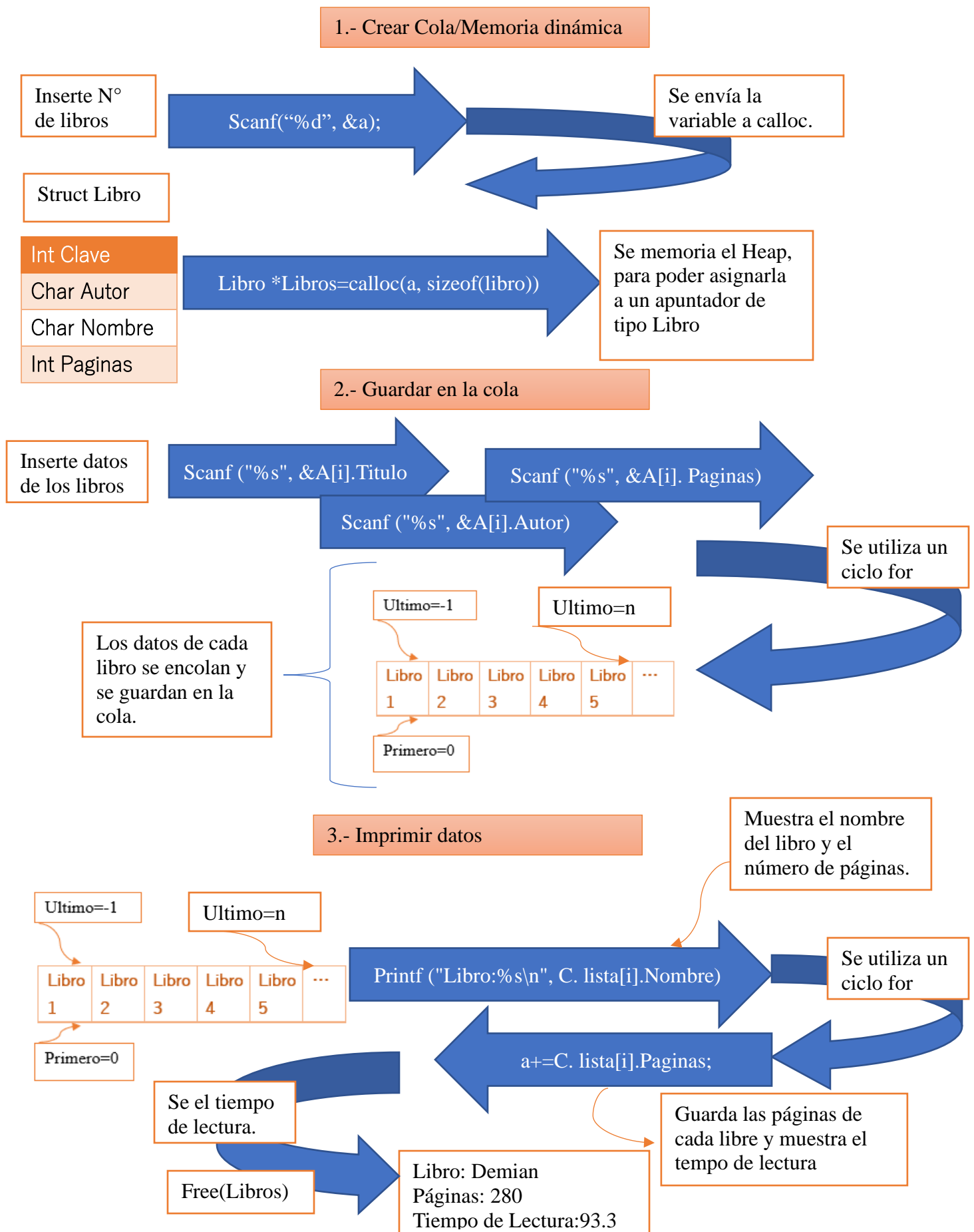
Imagen 2.

Para imprimir la sumatoria del tiempo total que se tardaría el usuario, en leer todos los libros insertados, lo que hice fue colocar una sumatoria, dentro del ciclo for que imprime los datos de cada libro, para que se almacenara en una variable las paginas de cada libro y por último multiplicarlas por un escalar, para obtener el tiempo de lectura total en minutos (Imagen 3).

```
printf("Paginas:%f, Tiempo de Lectura:%f\n",
a+=C.lista[i].Paginas;
```

Imagen 3

- Diagrama de funcionamiento



- Relación con teoría.

La teoría con la cual se puede resolver el problema, de acuerdo con a cómo lo resolví, se relaciona mas con las funciones de las colas, más con la función “encolar”, porque al modificar el tipo de dato de las funciones es que se volvió posible el guardado de los libros dentro de la cola. Y las demás instrucciones se acercan mas con los ciclos que facilitan el guardado de datos que el usuario introduce.

The screenshot displays a C++ IDE with two windows. The main window shows the source code for 'Ejercicio4.c', which implements a queue-based system for managing books. The code includes functions for inserting books (insertar), reading data (Leer), and calculating total reading time. The second window shows the program's output, where three books are entered: Joyce (230 pages), DuChard (300 pages), and Arturo (500 pages). The program then displays the total reading time for each book in minutes.

```
printf("Inserta el Titulo:\n");
scanf("%s",&A[i].Titulo);
printf("Insertar el Autor:\n");
scanf("%s",&A[i].Autor);
printf("Insertar N° de paginas:\n");
scanf("%f",&A[i].Paginas);
printf("Insertar la clave:\n");
scanf("%d",&A[i].Clave);
encolar(C,A[i]);
}

void Leer(Cola C,int z){
    int i,a;
    a=0;
    for(i=0;i<z;i++){
        printf("\nLos datos de los libro[%d]:\n",i+1);
        printf("Libro: %s\n",C.lista[i].Autor);
        printf("Paginas: %f, Tiempo de Lectura: %f minutos\n",C.lista[i].Paginas,C.lista[i].Clave);
        a+=C.lista[i].Paginas;
    }
    printf("Suma Total de lectura: %f",a*(1.0/3));
}
```

Inserta los libros a registrar:3

Datos del libro[1]
Inserta el Titulo:
Demian
Insertar el Autor:
Joyce
Insertar N° de paginas:
230
Insertar la clave:
567547

Datos del libro[2]
Inserta el Titulo:
Jean
Insertar el Autor:
DuChard
Insertar N° de paginas:
300
Insertar la clave:
4643644

Datos del libro[3]
Inserta el Titulo:
Detectives
Insertar el Autor:
Arturo
Insertar N° de paginas:
500
Insertar la clave:
54335234

Los datos de los libro[1]:
Libro:Joyce
Paginas:230.000000, Tiempo de Lectura:76.666667 minutos

Los datos de los libro[2]:
Libro:DuChard
Paginas:300.000000, Tiempo de Lectura:100.000000 minutos

Los datos de los libro[3]:
Libro:Arturo
Paginas:500.000000, Tiempo de Lectura:166.666667 minutos

Conclusiones.

En base al objetivo que se plantea al inicio de la practica puedo decir que aprendí bastante, comencé con muchas dudas la práctica, pero poco a poco las fui aclarando, tanto revisando apuntes como, haciendo los códigos de los ejercicios, a pesar de haberme atorado, trate de entender al máximo el funcionamiento de las estructuras cola y pila. Para finalizar puedo decir que se cumplió con el objetivo, pero me hubieran gustado ejemplos en la guía de la práctica, debido a que la teoría ayuda, pero siempre un ejemplo ilustra muchas cosas mejor.