



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: **Manuel Enrique Castañeda Castañeda**

Asignatura:

Estructura de Datos y Algoritmos I

Grupo:

12

No de Práctica(s):

05 . Estructuras de Datos Lineales: Pila & Cola

Integrante(s):

- **Lemus Ambrosio Aline Andrea**
- **Reyes Fuentes José Manuel**
- **Sánchez Alvirde Andrés Iván**
- **Wong Sánchez Yibran Lee**

*No. de Equipo de cómputo
empleado:*

-

No. de Lista o Brigada:

12

Semestre:

2021-2

Fecha de entrega:

2 de julio de 2021

Observaciones:

Calificación:

Estructura de Datos y Algoritmos I

Práctica 05

Estructuras de Datos Lineales: Pila & Cola

INTRODUCCIÓN

Durante esta práctica se estará usando los conceptos de pila y cola, con los cuales también veremos sus aplicaciones con los ejercicios dejados en esta práctica.

Comenzaremos hablando de las pilas de cómo son sus estructuras y su forma de uso.

Pila:

La pila la podemos catalogar como una (stack) es una estructura de datos, que consta de una serie de datos, en la cual las inserciones y eliminaciones se hacen por un extremo, llamado cima (top), de la pila. La estructura pila se conoce como LIFO (last-in, first-out, último en entrar, primero en salir), que significa "último elemento introducido, primero sacado".

Una pila al ser una lista puede almacenar en el campo de información cualquier tipo de valor (int, char, float, vector de caracteres, un objeto, o alguna otra variable).

Inicialmente la pila está vacía y decimos que el puntero raíz apunta a NULL (Si apunta a NULL decimos que no tiene una dirección de memoria, en realidad este valor NULL es el valor cero).

Las operaciones fundamentales introducir y eliminar se hacen por un extremo de la pila llamado cima (top).

Ejemplos de las pilas son:

•Navegador Web

-Se almacenan los sitios previamente visitados

-Cuando el usuario quiere regresar (presiona el botón de retroceso), simplemente se extrae la última dirección (pop) de la pila de sitios visitados.

• Editores de texto

– Los cambios efectuados se almacenan en una pila

– Usualmente implementada como arreglo

– Usuario puede deshacer los cambios mediante la operación "undo", la cual extraer el estado del texto antes del último cambio realizado.

La cola:

La cola podemos decir que son secuencias de elementos caracterizadas porque las operaciones de inserción y borrado se realizan sobre extremos opuestos de la secuencia. La inserción se produce en el "final" de la secuencia, mientras que el borrado se realiza en el otro extremo, el "inicio" de la secuencia.

Las restricciones definidas para una cola hacen que el primer elemento que se inserta en ella sea, igualmente, el primero en ser extraído de la estructura. Si una serie de elementos A, B, C, D, E se insertan en una cola en ese mismo orden, entonces los elementos irán saliendo de la cola en el orden en que entraron. Por esa razón, en ocasiones, las colas se conocen con el nombre de listas o secuencias FIFO (First In First Out, "el primero que entra es el primero que sale"), en la cual se puede hacer cualquier analogía donde se implementen como las colas para las tortillas o en algún comercio, así como llegan se van.

Algunas de las funciones u operaciones para la cola pueden ser: Inicializar la cola, añadir un elemento al final de la cola, eliminar el primer elemento de la cola, vaciar la cola y verificar el estado de la cola: Vacía / Llena.

Ejemplos:

– Trabajos enviados a una impresora

– Solicitudes a un servidor

– Clientes que quieren ser atendidos por una telefonista

DESARROLLO

1. Implementar en una pila con nodos, la verificación de que esté o no vacía

CÓDIGO FUENTE	EJECUCIÓN
<pre>//implementacion de pila #include <stdio.h> #include <stdlib.h> struct nodo{ int dato; nodo *siguiente; }; nodo *pila; //variable de tipo estructurada nodo, llamada pila void push();//void no devuelve nada int pop();//elimina el dato y le dice al usuario void imprimir(nodo *);//recibe un apuntador de tipo nodo void ver();//verifica si la pila esta vacia main(){ int op; do{ printf("Que operacion deseas realizar\n"); printf("1.-Insertar\n"); printf("2.-Eliminar\n"); printf("3.-Imprimir\n"); printf("4.-Saber si esta vacía\n"); printf("5.-Salir\n"); scanf("%d",&op); switch(op){ case 1:push();break; //push inserta los datos case 2:printf("Se elimino: %d",pop());break;//pop elimina case 3:imprimir(pila);break; case 4:ver();break; case 5:printf("ADIÓS"); } }while(op<5); } void push(){</pre>	 <pre>Que operacion deseas realizar 1.-Insertar 2.-Eliminar 3.-Imprimir 4.-Saber si la pila esta vacia 5.-Salir 4 La pila esta vacia Que operacion deseas realizar 1.-Insertar 2.-Eliminar 3.-Imprimir 4.-Saber si la pila esta vacia 5.-Salir 1 Ingresa el nuevo dato:5 Que operacion deseas realizar 1.-Insertar 2.-Eliminar 3.-Imprimir 4.-Saber si la pila esta vacia 5.-Salir</pre>

```

        nodo
*nuevo=(nodo*)malloc(sizeof(nodo)); //re
servamos el espacio necesario para
una pila
        nuevo->siguiente=pila;
        printf("Ingresa el nuevo dato:");
        scanf("%d",&nuevo->dato);
        pila = nuevo; //pila variable
auxiliar para guardar a nuevo
    }
int pop(){
    nodo *elim;
    int x;
    if(pila==NULL){ //NULL nos
pregunta si son identicos
        printf("La pila esta vacia");
        return 0;
    }else{
        x = pila->dato;
        elim = pila; //elim es el
ultimo elemento
        pila = pila->siguiente;
        free(elim); //libera el
espacio de memoria reservado
        return x; //indica dato
borrado
    }
}
void imprimir(nodo *){
    nodo *indice = pila;
    while(indice!=NULL){

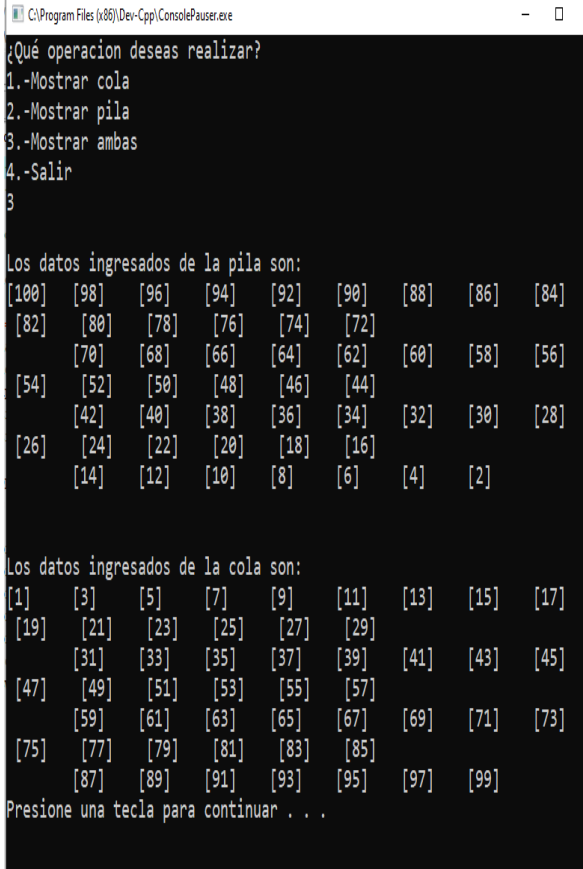
printf("[%d]\t",indice->dato); //imprime
los datos actuales
        indice = indice->siguiente;
//avanza los elementos de la pila
    }

    printf("\n");
}
void ver(){
    if(pila==NULL){ //NULL
nos pregunta si son identicos
        printf("La pila esta
vacía\n\n");
    }
    else{
        printf("La pila tiene

```

<pre> datos\n\n"); } } </pre>	
--	--

2. Recorrer los números del 1 al 100, los pares agregarlos a una pila y los impares a una cola

CÓDIGO FUENTE	EJECUCIÓN
<pre> #include <stdio.h> #include <stdlib.h> struct nodo{ int dato; nodo *siguiente; }; nodo *fin=NULL; nodo *pila; nodo *inicio=NULL; void pushp(); void mostrcol(); void mostrpil(); main(){ int op, op1; pushp(); do{ printf("%cQu%c operacion deseas realizar?\n", 168,130); printf("1.-Mostrar cola\n"); printf("2.-Mostrar pila\n"); printf("3.-Mostrar ambas\n"); printf("4.-Salir\n"); scanf("%d",&op); switch(op){ case 1:mostrcol(); system ("pause"); system ("cls"); break; case 2:mostrpil(); system ("pause"); system ("cls"); break; case 3: </pre>	 <pre> ¿Qué operacion deseas realizar? 1.-Mostrar cola 2.-Mostrar pila 3.-Mostrar ambas 4.-Salir 3 Los datos ingresados de la pila son: [100] [98] [96] [94] [92] [90] [88] [86] [84] [82] [80] [78] [76] [74] [72] [70] [68] [66] [64] [62] [60] [58] [56] [54] [52] [50] [48] [46] [44] [42] [40] [38] [36] [34] [32] [30] [28] [26] [24] [22] [20] [18] [16] [14] [12] [10] [8] [6] [4] [2] Los datos ingresados de la cola son: [1] [3] [5] [7] [9] [11] [13] [15] [17] [19] [21] [23] [25] [27] [29] [31] [33] [35] [37] [39] [41] [43] [45] [47] [49] [51] [53] [55] [57] [59] [61] [63] [65] [67] [69] [71] [73] [75] [77] [79] [81] [83] [85] [87] [89] [91] [93] [95] [97] [99] Presione una tecla para continuar . . . </pre>

```

        printf("\nLos datos
ingresados de la pila son:\n");
        mostrpil();
        printf("\n");
        printf("\nLos datos
ingresados de la cola son:\n");
        mostrcol();
        system("pause");
        system("cls");
        break;
        case 4:
        printf("Esperamos
que te hayas divertido en compu FI");
        break;
        default:
        printf("lamento
informarte que esa opci%cn, no es
valida\n", 162);
        break;
    }
}while(op<4);
}

void mostrpil(){
    nodo *indice = pila;
    while(indice!=NULL){

printf("[%d]\t",indice->dato);
        indice = indice->siguiente;
    }
    printf("\n");
}

void mostrcol(){
    nodo *indice=inicio;
    while(indice!=NULL){

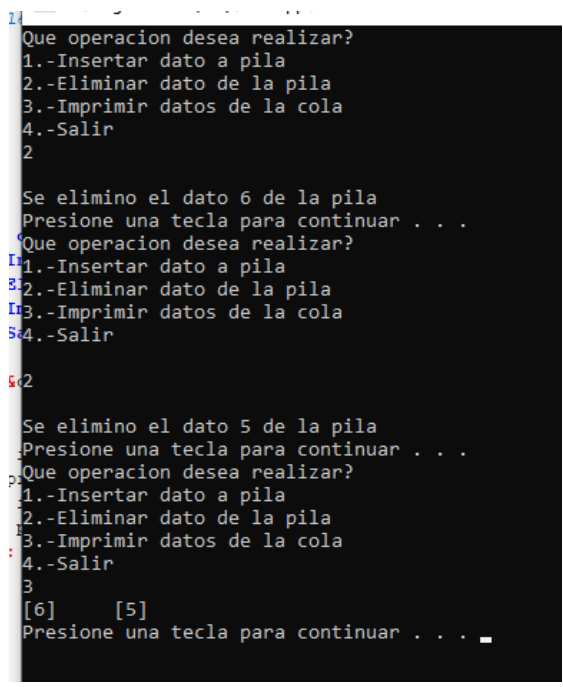
printf("[%d]\t",indice->dato);
        indice=indice->siguiente;
    }
    printf("\n");
}

void pushp(){ //aqui se llevaran todas
las operaciones para llenado
    int x;
    for (x=1;x<=100;x++){
        if(x%2==0 || x==0){
            nodo
*nuevo=(nodo*)malloc(sizeof(nodo));
            nuevo->siguiente=pila;

```

<pre> nuevo->dato=x; pila = nuevo; } else{ nodo *nuevo = (nodo*)malloc(sizeof(nodo)); nuevo->dato = x; nuevo->siguiente = NULL; if(inicio==NULL){ inicio=nuevo; fin=nuevo; } else{ fin->siguiente=nuevo; fin=nuevo; } } } } </pre>	
--	--

3. Ingresar datos a una pila, al extraer datos de esta pila, estos datos ingresaran a una cola

CÓDIGO FUENTE	EJECUCIÓN
<pre> #include <stdio.h> #include <stdlib.h> struct nodo{ int dato; nodo *siguiente; }; nodo *pila;//Variable de tipo estructura nodo, llamada pila nodo *final=NULL; nodo *inicial=NULL; void insertar(); int elimp(); void imprimir(); main(){ int op; do{ printf("Que operacion desea realizar?\n") ; printf("1.-Insertar dato a pila\n"); printf("2.-Eliminar dato de la pila\n"); </pre>	 <p>The screenshot shows the program's execution in a terminal. It displays a menu with four options: 1.-Insertar dato a pila, 2.-Eliminar dato de la pila, 3.-Imprimir datos de la cola, and 4.-Salir. The user enters '2' to remove a data item. The program then prompts for a key press to continue. This sequence is repeated once more, where the user again enters '2' to remove another data item. The terminal output shows the state of the stack and queue after each operation.</p>


```

printf("3.-Imprimir datos de
la cola\n");
printf("4.-Salir\n");

scanf("%d",&op);
switch(op)
{
    case 1: insertar();
    system("cls"); break;
    case 2:printf("\nSe
elimino el dato %d de la pila\n"
,elimp());system("\npause");break;
    case 3:
imprimir();system("cls"); break;
    case 4:
printf("\nEsperamos haberle ayudado
en compu Fi\n");break;
    default:
printf("\nOPCION NO VALIDA\n");
}
}while(op<4);
}

void insertar(){
    nodo
*nuevo=(nodo*)malloc(sizeof(nodo));//re
servar espacio para un elemento de pila
nuevo->siguiente=pila;
printf("\nIngresa el nuevo dato:
\n");
scanf("%d",&nuevo->dato);
pila=nuevo;//variable auxiliar
para guardar a nuevo
printf("Dato ingresado de manera
correcta\n\n");
system("pause");
}

int elimp(){
    nodo *elim;
    int x;
    if(pila==NULL){
        printf("La pila esta vacia");
        return 0;
    }else{
        x = pila->dato;
        elim = pila;
        pila = pila->siguiente;
        free(elim);
    }
}

```

```

        nodo *nuevo =
(nodo*)malloc(sizeof(nodo));
        nuevo->dato = x;
        nuevo->siguiente = NULL;
        if(inicial==NULL){
            inicial=nuevo;
            final=nuevo;
        }else{
final->siguiente=nuevo;
final=nuevo;
        }
        return x;
    }
}
void imprimir(){
    nodo *indice=inicial;
    if(inicial==NULL){
        printf("No hay datos de la
cola");
    }else{
        while(indice!=NULL){
printf("[%d]\t",indice->dato);
            indice=indice->siguiente;
        }
        printf("\n");
        system("pause");
    }
}

```

OBSERVACIONES (Individuales)

- **Lemus Ambrosio Aline Andrea**

Esta práctica se me hizo algo confusa por el uso de los nodos o apuntadores en para la pila y cola, afortunadamente este problema se pudo resolver con ayuda de mi equipo, ya que fue algo que tuve que buscar y analizar varias veces, dado que la pila y cola los intercala ríamos en dos de los ejercicios, sin duda esta práctica fue complicada, pese a que fueron solo tres ejercicios, si fue algo complicado para mi, con esto puedo saber que es necesario repasar estos temas y practicar más unir la pila y cola en un ejercicio; como dije antes con la ayuda de mi equipo y de algunos ejercicios hechos en clase, pudimos resolver la práctica.

- **Reyes Fuentes José Manuel**

Al estar realizando la práctica lo que más se complicó e ir entendiendo fue acerca de cómo implementar la pila y cola en el mismo ejercicio que ello lo tuvimos que hacer en algunos de los problema propuestos, sin duda fue algo que nos costó asimilar al igual acerca de cómo deberíamos declarar los nodos o más bien apuntadores donde se estarían guardando todas la variables e ir uniendo una con otra tanto como si fuera en cola o pila que tienen una forma diferente de impresión.

Con ayuda de programas hechos anteriormente se tomó la base que se tenía para ayudar a que fuera mucho más sencillo y claro sin olvidar de la ayuda de mis compañeros.

- **Sánchez Alvirde Andrés Iván**

Los ejercicios se me complicaron debido a que a la hora de implementar la cola y la pila me confundía mucho, esto porque son algo parecidos a la hora de guardar los datos pero se diferencian cuando toca eliminar algún dato de ambas. La declaración de nodos fue algo que también se dificultó al igual que el cómo ligar los datos con los otros, pero después de andar viendo los videos de clase, ver con detenimiento los ejercicios que se hicieron en clase al igual que ver las presentaciones de los compañeros que trataron los temas se aclararon las cosas, también fue la ayuda que nos dimos entre el equipo para acabar.

- **Wong Sánchez Yibran Lee**

Esta práctica me resultó con alto grado de complejidad pero con la ayuda de mi equipo logramos resolver diversas problemáticas, en un principio repasé diversos temas y conceptos que se llevaron a cabo en esta práctica, después de esto resultó más sencillo realizar los diversos programas.

CONCLUSIONES (Individuales)

- **Lemus Ambrosio Aline Andrea**

Esta práctica sin duda alguna me recuerda que debo de practicar más el unir la pila y la cola dado que esto fue lo que más se me complicó en la práctica, además del uso de los nodos o apuntadores en estas, por suerte se que tengo el apoyo de mis compañeros y profesor para resolver las dudas que me surjan y, pese a que la práctica se nos complicó, pudimos completarla de la mejor forma posible trabajando en equipo.

- **Reyes Fuentes José Manuel**

Estas dos formas de programar tienen su complejidad la manera que van entrelazados los datos una siguiente del otro y hacer que algunos otros cambien de lugar es un tanto difícil su comprensión, aun así al tener las bases y bien comprendido como es el flujo de datos es algo un tanto más “fácil”, lo pongo entre comillas porque sigue teniendo una complejidad que ha decir verdad si no se hubiera tenido el tiempo para hacerlo no se logra tan fácil.

Gracias a los errores fuimos aprendiendo y con ello aumentará nuestro conocimiento o tal vez mejor dicho nuestra forma de ponerlo en práctica.

- **Sánchez Alvirde Andrés Iván**

Concluyendo con esta práctica tuvo algo de dificultad por las cosas que me llegaron a confundir pero a pesar de eso se aprendieron muchas cosas de las pilas y colas para ir agregando datos e irlos ligando para almacenarlos. También se hizo repaso de las funciones para hacer los menús más simplificados.

- **Wong Sánchez Yibran Lee**

Puedo concluir de esta práctica que el uso de cola y pilas es muy útil para sintetizar código, también pudimos aprender a usar las funciones de las pilas y las colas como push, pop e imprimir y que funciones tienen estos al momento de crear códigos, además de reforzar el conocimiento de las funciones dentro de los códigos, a pesar de poder tener complicaciones y errores en los códigos.