



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor(a): Ing. Patricia Del Valle Morales

Asignatura: Estructura de Datos y Algoritmos I

Grupo: 02

No de Proyecto: 2. Laboratorios científicos UNAM

Integrante(s): Carrillo Alemán Luis Fernando

Chora López Adrián Santiago

Estrada Zacarias Aldo Axel

López Carreón Gabriel Augusto

*No. de lista o
brigada:* 06

Semestre: 2024-2

Fecha de entrega: 25/04/2024

Observaciones:

CALIFICACIÓN: _____

Estructuras de Datos y Algoritmos I

Ing. Patricia Del Valle Morales

EXAMEN-Proyecto 2

Estructuras dinámicas cola y cola circular, pila estática



El presente examen-proyecto cubre **los atributos de egreso** A1-CD2, A1-CD3, A2-CD1, A2-CD2 y A2-CD3 del **CACEI**:

- A1-CD2: expresar el fenómeno asociado mediante un modelo matemático adecuado
- A1-CD3: Aplicar la técnica correspondiente de la ingeniería para la resolución del problema
- A2-CD1: Simular las especificaciones del diseño requerido
- A2-CD2: determinar el proceso de diseño más adecuado para alcanzar los requerimientos indicados
- A2-CD3: Desarrollar proyectos que satisfacen las necesidades especificadas

El presente examen-proyecto cubre **los siguientes resultados de aprendizaje** para la certificación europea **ANECA**:

- Análisis en ingeniería. Capacidad de identificar, formular y resolver problemas de ingeniería. Capacidad de analizar productos, procesos y sistemas complejos.
- Aplicación práctica de la ingeniería.

I. Objetivo

Diseñar y desarrollar un programa de cómputo para el control y monitoreo de solicitudes de análisis de pruebas científicas de laboratorio aplicando las estructuras de datos básicas de la asignatura.

II. Restricciones:

- ✓ Desarrollar en equipo de exactamente el número de integrantes indicado
- ✓ Para que su proyecto sea evaluado deberá de cubrir los rubros obligatorios.
- ✓ La evaluación será individual
- ✓ La calificación individual se obtendrá a través de un examen individual sobre el código de tu examen o proyecto en el momento de presentarlo, obteniendo como calificación máxima la calificación de tu examen o proyecto.

III. Introducción

En un laboratorio científico del instituto de Química de la UNAM se reciben y procesan muestras de diferentes departamentos, facultades y dependencias privadas o gubernamentales. Es importante darle seguimiento al proceso y tener un control en tiempo real de las muestras solicitadas. El proceso se desarrolla en tres fases:

- ✓ Fase de recepción de muestras

- ✓ Fase de procesado y análisis
- ✓ Fase de reporte y entrega al departamento solicitante

Por ello se desea que se implemente un programa para el control y monitoreo de las muestras en tiempo real.

IV. Condiciones de diseño

A. Fase de recepción de muestras



A. Cola lineal dinámica: Muestras de entrada

Las muestras se registran y se depositan en una banda que los va acomodando en un contenedor de entrada conforme van llegando (**cola lineal dinámica- Nodos**) llamado contenedor de **muestras de entrada**; para que posteriormente sean analizadas por el laboratorio.

B. Fase de procesado y análisis



B. Cola circular dinámica: Muestras analizadas

El laboratorio cuenta con profesionistas para analizar las muestras, cada profesionista del laboratorio debe tomar una muestra a la vez, analizarla y una vez que la muestra está procesada depositarla en el contenedor (**cola circular dinámica-Nodos**) llamado contenedor de **muestras analizadas**.

C. Fase de reporte y entrega



C. Pila estática: Muestras por entregar

El laboratorio cuenta también con varios becarios que realizan funciones auxiliares, que son los que se encargan de realizar los registros e informes; toman las muestras de los contenedores y conforme se van liberando (de la **cola circular dinámica**) **Muestras analizadas** los apilan en el contenedor (**Pila estática-arreglo**) **Muestras por entregar**.

Finalmente, los repartidores del laboratorio son los que se encargan de entregar las muestras a los departamentos. Ellos toman las muestras del contenedor (**Pila estática-arreglo**) Muestras **por entregar** y reportan en el sistema que se encarga de imprimir a la pantalla el siguiente mensaje:

La muestra ##### se entregó al departamento ###

V. Simulación

1. Implementarlo con estructura de datos:
 - a. Cola simplemente enlazada,

ColaLSE.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "colaLSE.h"
4
5  Cola *crearCola(){
6      Cola *nuevaCola;
7      //CREACION DINAMICA DE LA ESTRUCTURA COLA
8      nuevaCola = (Cola *)malloc(1*sizeof(Cola));
9      if (nuevaCola == NULL){
10         printf("Error: Espacio insuficiente...");
11         exit(0);
12     }
13     //INICIALIZANDO h y t
14     nuevaCola->h = NULL;
15     nuevaCola->t = NULL;
16     return nuevaCola;
17 }
18
19 void insertar(Cola *cola, Muestra dato){
20     NodoC *nuevoNodo;
21     //CREA EL NODO
22     nuevoNodo = (NodoC *)malloc(sizeof(NodoC));
23     if (nuevoNodo == NULL){
24         printf("Error: memoria insuficiente...");
25         exit(0);
26     }
```

```

27 //1.ASIGNA VALORES AL NODO
28 nuevoNodo->info = dato;
29 nuevoNodo->sig = NULL;
30 //2.INSERTA EL NODO EN LA COLA
31 if (colaVacia(*cola))
32     cola->h = cola->t = nuevoNodo;
33 else {
34     cola->t->sig = nuevoNodo;
35     cola->t = nuevoNodo;
36 }
37 }
38
39 void listarCola(Cola cola){
40     NodoC *q;
41     printf("\n");
42     printf("\n===== \nMUESTRAS DE
ENTRADA\n===== \n");
43     if (colaVacia(cola))
44         printf("\nNo hay datos en la fila...\n");
45     else{
46         for(q = cola.h; q != NULL; q = q->sig)
47             listarMuestra(q->info);
48     }
49     printf("\n");
50 }
51
52 int colaVacia(Cola cola){
53     return cola.h == NULL;
54 }
55
56 Muestra borrar(Cola *cola){
57     Muestra dato;
58     NodoC *q = cola->h;
59     //BORRA EL DATO
60     if (!colaVacia(*cola)){
61         if (cola->h == cola->t)
62             cola->h = cola->t = NULL;
63         else
64             cola->h = cola->h->sig;
65         dato = q->info; // EXTRAER LA INFORMACIÓN
66         q->sig = NULL; // DESENLACE DE LA VARIABLE Q
67         free(q); // LIBERA LA MEMORIA
68     }
69     else
70         printf("\nNo hay datos registrados...");
71     return dato;
72 }
73
74 void inicializarCola(Cola *cola){
75     NodoC *q = cola->h;
76     if (!colaVacia(*cola)){
77         //LIBERA MEMORIA
78         while (q != NULL){
79             cola->h = cola->h->sig;

```

```

80     q->sig = NULL;
81     free(q); //LIBERA LA MEMORIA
82     q = cola->h;
83 }
84 //INICIALIZA APUNTADES COLA
85 cola->h = cola->t = NULL;
86 }
87 }
88
89 int numeroElementos(Cola cola){
90     int num = 0;
91     NodoC *q;
92     if (!colaVacía(cola)){
93         for(q = cola.h; q != NULL; q = q->sig)
94             num++;
95     }
96     return num;
97 }

```

ColaLSE.h

```

1  #ifndef COLADINAMICA_H_INCLUDED
2  #define COLADINAMICA_H_INCLUDED
3
4  #include "muestra.h"
5
6  typedef struct nodoC{
7      Muestra info;
8      struct nodoC *sig; //APUNTADE AL SIG NODO DE LA COLA
9  }NodoC;
10
11  typedef struct {
12      NodoC *h,*t;
13  } Cola;
14
15  //FUNCIONES
16  void insertar(Cola *cola, Muestra dato);
17  Muestra borrar(Cola *cola);
18  //FUNCIONES AUXILIARES
19  Cola *crearCola();
20  int colaVacía(Cola cola);
21  void inicializarCola(Cola *cola);
22  void listarCola(Cola cola);
23  int numeroElementos(Cola cola);
24
25  #endif // COLADINAMICA_H_INCLUDED

```

b. Cola circular simplemente enlazada

ColaCircularLSE.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "colaCircularLSE.h"
4  // Funciones
5  ColaCircular *crearColaCircular(){
6      ColaCircular *cc;
7      cc = (ColaCircular *)malloc(sizeof(ColaCircular));
8      if(cc == NULL){
9          printf("\nError al crear la cola...\n");
10         exit(0);
11     }
12     cc->h = NULL;
13     cc->t = NULL;
14     return cc;
15 }
16
17 int vacia(ColaCircular colaCir){
18     return colaCir.h == NULL;
19 }
20
21 void Insertar(ColaCircular *colaCir, Muestra dato){
22     //CREAR NODO
23     NodoCC *nuevo = (NodoCC *)malloc(sizeof(NodoCC));
24     if (nuevo == NULL){
25         printf("\nError al crear el nodo...");
26         exit(0);
27     }
28     nuevo->info = dato;
29     nuevo->sig = colaCir->h; //LIGADURA CIRCULAR
30 }
```



```

31 //INSERTAR NODO
32 if(vacia(*colaCir)){
33     colaCir->h = nuevo;
34     colaCir->t = nuevo;
35     colaCir->t->sig = colaCir->h; //LIGADURA CIRCULAR
36     return;
37 }
38 colaCir->t->sig = nuevo;
39 colaCir->t = nuevo;
40 }
41
42 void Listar(ColaCircular colaCir){
43     NodoCC *q;
44     q = colaCir.h;
45     printf("\n");
46     printf("\n=====\\nMUESTRAS ANALIZADAS\\n=====\\n");
47     if (q != NULL) {
48         do {
49             listarMuestra(q->info);
50             q = q->sig;
51         }while(q != colaCir.h);
52         printf("\n");
53     }
54     else
55         printf("\\nNo hay datos...\\n");
56 }
57
58 Muestra Borrar(ColaCircular *colaCir){
59     Muestra dato;
60     NodoCC *q = colaCir->h;
61     if(! vacia(*colaCir)) {
62         dato = q->info;
63         if (colaCir->h == colaCir->t)
64             colaCir->h = colaCir->t = NULL;
65         else {
66             colaCir->h = colaCir->h->sig;
67             colaCir->t->sig = colaCir->h;
68         }
69         q->sig = NULL;
70         free(q);
71     }
72     else
73         printf("\\nNo hay datos registrados...\\n");
74     return dato;
75 }
76
77 void inicializarColaCircular(ColaCircular *cc){
78     NodoCC *q = cc->h;
79     if (!vacia(*cc)){
80         //LIBERA MEMORIA
81         while (q != cc->t->sig){
82             cc->h = cc->h->sig;
83             q->sig = NULL;
84             free(q); //LIBERA LA MEMORIA
85             q = cc->h;
86         }
87         //INICIALIZA APUNTADES COLA
88         cc->h = cc->t = NULL;
89     }
90 }
91
92 int elementos(ColaCircular cc){
93     NodoCC *q = cc.h;
94     int n = 0;

```

```

95     printf("\n");
96     if (q != NULL) {
97         do {
98             n++;
99             q = q->sig;
100         }while(q != cc.h);
101     }
102     return n;
103 }

```

ColaCircularLSE.h

```

1  #ifndef COLACIRCULARLSE_H_INCLUDED
2  #define COLACIRCULARLSE_H_INCLUDED
3
4  #include "muestra.h"
5
6  typedef struct nodoCC NodoCC;
7  struct nodoCC{
8      Muestra info;
9      NodoCC *sig;
10 };
11
12 typedef struct{
13     NodoCC *h, *t;
14 } ColaCircular;
15
16 ColaCircular *crearColaCircular();
17 int vacia(ColaCircular colaCir);
18 void Insertar(ColaCircular *colaCir, Muestra dato);
19 void Listar(ColaCircular colaCir);
20 Muestra Borrar (ColaCircular *colaCir);
21 void inicializarColaCircular(ColaCircular *cc);
22 int elemto(ColaCircular cc);
23
24
25 #endif // COLACIRCULARLSE_H_INCLUDED

```

c. Pila estática de tipo de datos muestra

Pila.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "pila.h"
4
5 Pila *crearPila(int n){
6     Pila *nuevaPila;
7     //1. crear la pila y el arreglo de la pila
8     nuevaPila = (Pila *)malloc(1 *sizeof(Pila)); //CREA LA ESTRUCTURA PILA CON TODOS SUS ELEMENTOS
9     if(nuevaPila == NULL){
10         printf("Error: No hay espacio...");
11         exit(0);
12     }
13     nuevaPila->arrPila = (Muestra *)calloc(n , sizeof(Muestra)); //CREANDO EL ARREGLO DE LA PILA
14     if(nuevaPila->arrPila == NULL){
15         printf("Error: No hay espacio...");
16         exit(0);
17     }
18     //2. INICIALIZA VALORES
19     nuevaPila->tope = -1;
20     nuevaPila->max = n;
21     return nuevaPila;
22 }
23
24 int pilaLlena(Pila *pila){
25     return (pila->tope >= pila->max-1);
26 }
```

```

27
28 void push(Muestra dato,Pila *pila){
29     pila->tope++;
30     pila->arrPila[pila->tope] = dato;
31 }
32
33 void listarPila(Pila pila){
34     printf("\n===== \nMUESTRAS POR ENTREGAR\n===== \n");
35     for (int i= pila.tope; i >= 0; i--)
36         listarMuestra(pila.arrPila[i]);
37 }
38
39 int pilaVacía(Pila *pila){
40     return (pila->tope == -1);
41 }
42
43 Muestra pop(Pila *pila){
44     Muestra aux = pila->arrPila[pila->tope];
45     pila->tope--;
46     return aux;
47 }
48
49 void inicializarPila(Pila *pila){
50     pila->tope = -1;
51 }
52
53 void liberarMemoriaPila(Pila *pila){
54     free(pila->arrPila);
55     free(pila);
56     pila = NULL;
57 }
58

```

```
1  #ifndef PILA_H_INCLUDED
2  #define PILA_H_INCLUDED
3
4  #include "muestra.h"
5
6  typedef struct {
7      Muestra *arrPila;
8      int max;
9      int tope;
10 } Pila;
11
12 Pila *crearPila(int max);
13 int pilaLLena(Pila *pila);
14 void push(Muestra dato, Pila *pila);
15 void listarPila(Pila pila);
16 int pilaVacía(Pila *pila);
17 Muestra pop(Pila *pila);
18 void inicializarPila(Pila *pila);
19 void liberarMemoriaPila(Pila *pila);
20
21 #endif // PILA_H_INCLUDED
```

2. Datos del objeto muestra:
 - a. Clave (numérica entera)
 - b. Nombre (cadena de caracteres)
 - c. Nombre del departamento al que pertenece la muestra

muestra.h

```
1  #ifndef MUESTRA_H_INCLUDED
2  #define MUESTRA_H_INCLUDED
3
4  #define MAX 100000
5  #define MAX_TEXTO 50
6
7  // estructuras
8  typedef struct{
9      int clave;
10     char *nombre;
11     char *nombreDep;
12 } Muestra;
13
14 // prototipos
15 Muestra *crearMuestra();
16 void registrarMuestra(Muestra *unaMuestra);
17 void listarMuestra(Muestra unaMuestra);
18
19 #endif // MUESTRA_H_INCLUDED
```

muestra.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <time.h>
5
6  #include "muestra.h"
7
8  // variable global para llenar las muestras
9  char nombresMuestras[20][MAX_TEXTO] = {"Sangre", "Orina", "Tejidos", "Saliva", "Pelo", "Huellas dactilares",
10 "Fragmentos de vidrio", "Residuos de Polvora",
11 "Tintas y papel", "Fibras textiles", "Pinturas y pigmentos", "Tierra y
12 suciedad", "Cenizas y residuos de incendios",
13 "Fragmentos de explosivos", "Armas de fuego y municiones", "Vidrios y
14 ventanas", "Muestras de suelo y agua",
15 "Muestras de alimentos y bebidas", "Objetos personales", "Productos
16 químicos y sustancias sintéticas"};
17 char nombresDepartamentos[10][MAX_TEXTO] = {"Toxicología Forense", " Identificación de Drogas", "Análisis de
18 Fluidos Corporales", "Química de Explosivos",
19 "Análisis de Incendios", "Serología Forense", "Balística Forense",
20 "Química de Documentos",
21 "Análisis de Pinturas y Fibras", "Química Forense Ambiental"};
22
23 Muestra *crearMuestra(){
24     Muestra *nuevaMuestra = (Muestra *) calloc(1, sizeof(Muestra));
25     if (nuevaMuestra == NULL) {
26         printf("\nError: No se pudo asignar memoria para el arreglo de libros.\n");
27         exit(0);
28     }
29     nuevaMuestra->nombre = (char *) malloc(MAX_TEXTO * sizeof(char));
30     nuevaMuestra->nombreDep = (char *) malloc(MAX_TEXTO * sizeof(char));
31     return nuevaMuestra;
32 }
33
34 void registrarMuestra(Muestra *unaMuestra){
35     srand(time(NULL));
36     unaMuestra->clave = rand()%1000;
37     strcpy(unaMuestra->nombre, nombresMuestras[rand()%20]);
38     strcpy(unaMuestra->nombreDep, nombresDepartamentos[rand()%10]);
39 }
40
41 void listarMuestra(Muestra unaMuestra){
42     printf("\nClave: %d\n", unaMuestra.clave);
43     printf("Nombre: %s\n", unaMuestra.nombre);
44     printf("Nombre del departamento: %s\n", unaMuestra.nombreDep);
45 }
```

3. La generación de muestras de entrada es aleatoria, si el número aleatorio está dentro del 0-50 se generará una nueva muestra, en caso contrario si el número aleatorio es de 51 a 100 NO se genera muestra.
Deberá de imprimir a pantalla:
 - a. Si llega o no una muestra e imprimir los datos de la muestra generada
 - b. Imprimir el contenido del contenedor de muestras de entrada ya con la muestra registrada

laboratorio.h

```
1  #ifndef LABORATORIO_H_INCLUDED
2  #define LABORATORIO_H_INCLUDED
3
4  #include "muestra.h"
5  #include "colaLSE.h"
6  #include "colaCircularLSE.h"
7  #include "pila.h"
8
9  // m (Muestras), me (Muestras de Entrada), ma (Muestras Analizadas), mp (Muestras por Entregar)
10
11 // prototipos
12 void recepcionMuestras(Cola *me, ColaCircular *ma, Pila *mpe);
13 void procesadoAnálisis(Cola *me, ColaCircular *ma, Pila *mpe);
14 void reporteEntrega(Cola *me, ColaCircular *ma, Pila *mpe);
15
16 #endif // LABORATORIO_H_INCLUDED
```


Laboratorio.c

- Fase Recepción de Muestras:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <time.h>
5  #include "laboratorio.h"
6
7  void recepcionMuestras(Cola *me, ColaCircular *ma, Pila *mpe){
8      int n = rand()%100;
9      system("cls");
10     printf("\n\n////////////////////////////////\n// Recepcion de Muestras //\n////////////////////////////////\n\n");
11     if(n <= 50){
12         for(int i = 0; i < 5; i++){
13             Muestra *nuevaMuestra = crearMuestra();
14             registrarMuestra(nuevaMuestra);
15             listarMuestra(*nuevaMuestra);
16             insertar(me, *nuevaMuestra);
17             system("pause");
18         }
19         system("cls");
20         listarCola(*me);
21         system("pause");
22     }
23     else{
24         printf("\nNo se han registrado muestras.\n");
25         system("pause");
26     }
27     system("cls");
28     return procesadoAnalisis(me, ma, mpe);
29 }
```

Corrida 1

=====
MUESTRAS DE ENTRADA
=====

Clave: 976
Nombre: Pinturas y pigmentos
Nombre del departamento: Toxicología Forense

Clave: 983
Nombre: Fragmentos de vidrio
Nombre del departamento: Análisis de Pinturas y Fibras

Clave: 986
Nombre: Sangre
Nombre del departamento: Toxicología Forense

Clave: 989
Nombre: Tejidos
Nombre del departamento: Análisis de Pinturas y Fibras

Clave: 993
Nombre: Vidrios y ventanas
Nombre del departamento: Análisis de Pinturas y Fibras

Presione una tecla para continuar . . . █

```
////////////////////////////////////////  
// Recepcion de Muestras //  
////////////////////////////////////////
```

```
No se han registrado muestras.  
Presione una tecla para continuar . . .
```

=====
MUESTRAS DE ENTRADA
=====

Clave: 761

Nombre: Fragmentos de explosivos

Nombre del departamento: Serología Forense

Clave: 787

Nombre: Tejidos

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 794

Nombre: Muestras de suelo y agua

Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █

4. Para simular que los profesionistas del laboratorio extraen muestras del contenedor de entrada y las analizan: se generará un número aleatorio entre 0 y el número máximo de muestras en el contenedor, este número representa el número de muestras que se extraerán para analizarlas y pasarán (insertarán) a la siguiente cola Muestras analizadas. Deberá de imprimir a pantalla:
 - a. Número de muestras que se analizarán
 - b. Imprimir el contenido del contenedor de muestras de entrada para observar las muestras pendientes por analizar.
 - c. La clave de la muestra en proceso con el siguiente mensaje.
Ejemplo: Procesando muestra 302
 - d. Después de imprimir el mensaje anterior deberá de insertarse la muestra en el contenedor de muestras por entregar.
 - e. Deberá de listar el contenedor de muestras por entregar para observar que se registró correctamente las muestra procesada y analizada.

Laboratorio.c

- Fase de Procesado y Análisis:

```

31 void procesadoAnálisis(Cola *me, ColaCircular *ma, Pila *mpe){
32     Muestra unaMuestra, muestrasAnalizadas;
33     int e = numeroElementos(*me), n;
34     n = rand()%e;
35     printf("\n\n////////////////////////////////\n// PROCESADO Y ANALISIS //\n////////////////////////////////\n\n");
36
37     printf("\nEl numero de muestras que se analizaran es %i\n\n", n);
38     system("pause");
39
40     for(int i = 0; i < n; i++){
41         listarCola(*me);
42         unaMuestra = borrar(me);
43         system("pause");
44         system("cls");
45
46         printf("\nProcesando la muestra con clave %i\n", unaMuestra.clave);
47         Insertar(ma, unaMuestra);
48
49         Listar(*ma);
50         system("pause");
51         system("cls");
52     }
53     for(int j = 0; j < n; j++){
54         muestrasAnalizadas = Borrar(ma);
55         push(muestrasAnalizadas, mpe);
56     }
57     system("cls");
58     listarPila(*mpe);
59     system("pause");
60     system("cls");
61     return reporteEntrega(me, ma, mpe);
62 }

```

Corrida 1

```
=====
MUESTRAS DE ENTRADA
=====
```

```
Clave: 976
Nombre: Pinturas y pigmentos
Nombre del departamento: Toxicología Forense
```

```
Clave: 983
Nombre: Fragmentos de vidrio
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Clave: 986
Nombre: Sangre
Nombre del departamento: Toxicología Forense
```

```
Clave: 989
Nombre: Tejidos
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Clave: 993
Nombre: Vidrios y ventanas
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Presione una tecla para continuar . . . █
```

```
////////////////////////////////////
// PROCESADO Y ANALISIS //
////////////////////////////////////
```

```
El numero de muestras que se analizaran es 2
```

```
Presione una tecla para continuar . . . █
```



```
=====
MUESTRAS DE ENTRADA
=====
```

```
Clave: 976
Nombre: Pinturas y pigmentos
Nombre del departamento: Toxicología Forense
```

```
Clave: 983
Nombre: Fragmentos de vidrio
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Clave: 986
Nombre: Sangre
Nombre del departamento: Toxicología Forense
```

```
Clave: 989
Nombre: Tejidos
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Clave: 993
Nombre: Vidrios y ventanas
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Presione una tecla para continuar . . . █
```

```
Procesando la muestra con clave 976
```

```
=====
MUESTRAS ANALIZADAS
=====
```

```
Clave: 976
Nombre: Pinturas y pigmentos
Nombre del departamento: Toxicología Forense
```

```
Presione una tecla para continuar . . . █
```

```
=====
MUESTRAS DE ENTRADA
=====
```

```
Clave: 983
Nombre: Fragmentos de vidrio
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Clave: 986
Nombre: Sangre
Nombre del departamento: Toxicología Forense
```

```
Clave: 989
Nombre: Tejidos
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Clave: 993
Nombre: Vidrios y ventanas
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Presione una tecla para continuar . . . █
```

```
Procesando la muestra con clave 983
```

```
=====
MUESTRAS ANALIZADAS
=====
```

```
Clave: 976
Nombre: Pinturas y pigmentos
Nombre del departamento: Toxicología Forense
```

```
Clave: 983
Nombre: Fragmentos de vidrio
Nombre del departamento: Análisis de Pinturas y Fibras
```

```
Presione una tecla para continuar . . . █
```



```
=====
MUESTRAS POR ENTREGAR
=====
```

Clave: 983

Nombre: Fragmentos de vidrio

Nombre del departamento: Análisis de Pinturas y Fibras

Clave: 976

Nombre: Pinturas y pigmentos

Nombre del departamento: Toxicología Forense

Presione una tecla para continuar . . . █

Corrida 2

```
=====
MUESTRAS DE ENTRADA
=====
```

```
Clave: 814
Nombre: Fragmentos de explosivos
Nombre del departamento: Balística Forense
```

```
Clave: 820
Nombre: Fibras textiles
Nombre del departamento: Serología Forense
```

```
Clave: 837
Nombre: Armas de fuego y municiones
Nombre del departamento: Análisis de Incendios
```

```
Clave: 863
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales
```

```
Clave: 866
Nombre: Pelo
Nombre del departamento: Análisis de Fluidos Corporales
```

```
Presione una tecla para continuar . . . █
```

```
////////////////////////////////////
// PROCESADO Y ANALISIS //
////////////////////////////////////
```

```
El numero de muestras que se analizaran es 0
```

```
Presione una tecla para continuar . . . █
```

```
=====
MUESTRAS POR ENTREGAR
=====
```

```
Presione una tecla para continuar . . . █
```

Corrida 3

```
=====
MUESTRAS DE ENTRADA
=====
```

Clave: 761

Nombre: Fragmentos de explosivos

Nombre del departamento: Serología Forense

Clave: 787

Nombre: Tejidos

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 794

Nombre: Muestras de suelo y agua

Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █

```
////////////////////////////////////
// PROCESADO Y ANALISIS //
////////////////////////////////////
```

El numero de muestras que se analizaran es 4

Presione una tecla para continuar . . . █

=====

MUESTRAS DE ENTRADA

=====

Clave: 761

Nombre: Fragmentos de explosivos

Nombre del departamento: Serología Forense

Clave: 787

Nombre: Tejidos

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 794

Nombre: Muestras de suelo y agua

Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █

Procesando la muestra con clave 761

=====
MUESTRAS ANALIZADAS
=====

Clave: 761
Nombre: Fragmentos de explosivos
Nombre del departamento: Serología Forense

Presione una tecla para continuar . . . █

=====
MUESTRAS DE ENTRADA
=====

Clave: 787
Nombre: Tejidos
Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales

Clave: 794
Nombre: Muestras de suelo y agua
Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █

Procesando la muestra con clave 787

=====
MUESTRAS ANALIZADAS
=====

Clave: 761
Nombre: Fragmentos de explosivos
Nombre del departamento: Serología Forense

Clave: 787
Nombre: Tejidos
Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █

=====
MUESTRAS DE ENTRADA
=====

Clave: 790
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales

Clave: 794
Nombre: Muestras de suelo y agua
Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █

Procesando la muestra con clave 790

=====
MUESTRAS ANALIZADAS
=====

Clave: 761
Nombre: Fragmentos de explosivos
Nombre del departamento: Serología Forense

Clave: 787
Nombre: Tejidos
Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █

=====
MUESTRAS DE ENTRADA
=====

Clave: 790
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales

Clave: 794
Nombre: Muestras de suelo y agua
Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █

Procesando la muestra con clave 790

=====
MUESTRAS ANALIZADAS
=====

Clave: 761

Nombre: Fragmentos de explosivos

Nombre del departamento: Serología Forense

Clave: 787

Nombre: Tejidos

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Presione una tecla para continuar . . . █


```
=====
MUESTRAS POR ENTREGAR
=====
```

```
Clave: 790
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales
```

```
Clave: 790
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales
```

```
Clave: 787
Nombre: Tejidos
Nombre del departamento: Análisis de Fluidos Corporales
```

```
Clave: 761
Nombre: Fragmentos de explosivos
Nombre del departamento: Serología Forense
Presione una tecla para continuar . . . █
```

5. Para simular que los repartidores del laboratorio extraen muestras del contenedor de “muestras por entregar”: se generará un número aleatorio entre 0 y el número máximo de muestras almacenadas en el último repositorio (fila de salida), este número representa el número de muestras que se extraerán para entregar los resultados a los departamentos. Deberá de imprimir a pantalla:
- Número de muestras a entregar
 - Imprimir el contenido del contenedor de muestras por entregar para observar las muestras pendientes por entregar.
 - La clave de la muestra en proceso con el siguiente mensaje.
Ejemplo: La muestra ##### se entregó al departamento ###

Laboratorio.c

- Fase de Reporte y Entrega

```
64 void reporteEntrega(Cola *me, ColaCircular *ma, Pila *mpe){
65     Muestra otraMuestra;
66     int e = mpe->tope, n;
67     n = rand() % e;
68     printf("\n\n////////////////////////////////\n// REPORTE Y ENTREGA //\n////////////////////////////////\n\n");
69
70     printf("\nEl numero de muestras a entregar es %i\n", n);
71     system("pause");
72
73     for(int i = 0; i < n; i++){
74         listarPila(*mpe);
75         otraMuestra = pop(mpe);
76         system("pause");
77         system("cls");
78
79         printf("\nLa muestra %s de clave %i se devolvio al departamento %s\n",
80             otraMuestra.nombre, otraMuestra.clave, otraMuestra.nombreDep);
81         system("pause");
82         system("cls");
83     }
84     system("cls");
85     listarCola(*me);
86     Listar(*ma);
87     listarPila(*mpe);
88     system("pause");
89     system("cls");
90     return recepcionMuestras(me, ma, mpe);
91 }
```

Corrida 1

```
////////////////////  
// REPORTE Y ENTREGA //  
////////////////////
```

El numero de muestras a entregar es 0
Presione una tecla para continuar . . . █

Corrida 2

```
////////////////////  
// REPORTE Y ENTREGA //  
////////////////////
```

El numero de muestras a entregar es 0
Presione una tecla para continuar . . . █

Corrida 3

```
////////////////////  
// REPORTE Y ENTREGA //  
////////////////////
```

El numero de muestras a entregar es 2
Presione una tecla para continuar . . . █

=====

MUESTRAS POR ENTREGAR

=====

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 787

Nombre: Tejidos

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 761

Nombre: Fragmentos de explosivos

Nombre del departamento: Serología Forense

Presione una tecla para continuar . . . █

La muestra Saliva de clave 790 se devolvió al departamento Análisis de Fluidos Corporales
Presione una tecla para continuar . . . █

```
=====
MUESTRAS POR ENTREGAR
=====
```

Clave: 790

Nombre: Saliva

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 787

Nombre: Tejidos

Nombre del departamento: Análisis de Fluidos Corporales

Clave: 761

Nombre: Fragmentos de explosivos

Nombre del departamento: Serología Forense

Presione una tecla para continuar . . . █

La muestra Saliva de clave 790 se devolvió al departamento Análisis de Fluidos Corporales
Presione una tecla para continuar . . . █

6. Cada vez que se agreguen muestras o se eliminen muestras en cualquiera de los contenedores “Muestras de entrada”, “Muestras analizadas” y la pila “Muestras por entregar” deberán de imprimirse las tres para que el usuario del sistema pueda monitorear el estado de cada muestra de laboratorio.

Corrida 1

```
=====
MUESTRAS DE ENTRADA
=====

Clave: 986
Nombre: Sangre
Nombre del departamento: Toxicología Forense

Clave: 989
Nombre: Tejidos
Nombre del departamento: Análisis de Pinturas y Fibras

Clave: 993
Nombre: Vidrios y ventanas
Nombre del departamento: Análisis de Pinturas y Fibras


=====
MUESTRAS ANALIZADAS
=====

No hay datos...


=====
MUESTRAS POR ENTREGAR
=====

Clave: 983
Nombre: Fragmentos de vidrio
Nombre del departamento: Análisis de Pinturas y Fibras

Clave: 976
Nombre: Pinturas y pigmentos
Nombre del departamento: Toxicología Forense
Presione una tecla para continuar . . . █
```


Corrida 2

```
=====
MUESTRAS DE ENTRADA
=====
```

```
Clave: 814
Nombre: Fragmentos de explosivos
Nombre del departamento: Balística Forense
```

```
Clave: 820
Nombre: Fibras textiles
Nombre del departamento: Serología Forense
```

```
Clave: 837
Nombre: Armas de fuego y municiones
Nombre del departamento: Análisis de Incendios
```

```
Clave: 863
Nombre: Saliva
Nombre del departamento: Análisis de Fluidos Corporales
```

```
Clave: 866
Nombre: Pelo
Nombre del departamento: Análisis de Fluidos Corporales
```

```
=====
MUESTRAS ANALIZADAS
=====
```

No hay datos...

```
=====
MUESTRAS POR ENTREGAR
=====
```

Presione una tecla para continuar . . . █

Corrida 3

```
=====
MUESTRAS DE ENTRADA
=====
```

```
Clave: 794
Nombre: Muestras de suelo y agua
Nombre del departamento: Análisis de Fluidos Corporales
```

```
=====
MUESTRAS ANALIZADAS
=====
```

```
No hay datos...
```

```
=====
MUESTRAS POR ENTREGAR
=====
```

```
Clave: 787
Nombre: Tejidos
Nombre del departamento: Análisis de Fluidos Corporales
```

```
Clave: 761
Nombre: Fragmentos de explosivos
Nombre del departamento: Serología Forense
Presione una tecla para continuar . . . █
```


7. El programa se corre infinitamente.

main.c

- Ciclo infinito por medio de la llamada a las funciones

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #include "laboratorio.h"
5
6 int main(void){
7     // variables locales
8     Cola *muestrasDeEntrada = crearCola();
9     ColaCircular *muestrasAnalizadas = crearColaCircular();
10    Pila *muestrasPorEntregar = crearPila(MAX);
11
12    // codigo principal
13    printf("\n\n\t\t===== \n\t\tLaboratorio Quimico
Forense\n\t\t===== \n\n");
14    system("pause");
15    recepcionMuestras(muestrasDeEntrada, muestrasAnalizadas, muestrasPorEntregar);
16    return 0;
17 }
```

```
=====
Laboratorio Químico Forense
=====

Presione una tecla para continuar . . .
```

CONCLUSIONES

Carrillo Alemán Luis Fernando:

Durante este proyecto fortalecí mis conocimientos de lo que son las estructuras dinámicas y estáticas y como aplicarlas a la resolución de este programa, el cual fue el algoritmo para un laboratorio que constó de 3 etapas, las cuales cada una tuvo su dificultad, pero que al final se resolvieron gracias al trabajo en equipo y al esfuerzo de cada integrante. Tengo que decir, que el proyecto anterior se me había dificultado bastante debido a que no sabía cómo implementar las colas y pilas de manera correcta, sin embargo, al repasar y estudiar, este se me hizo bastante más fácil, ya que además de las plantillas que ya teníamos debido a las practicas, también comprendí teórica y prácticamente el uso de estas estructuras, como lo fueron la cola circular, cola con listas, y pila.

Realmente tuve pocos percances a la hora del desarrollo de este proyecto, ya que como lo mencione antes, estude y comprendí mejor los temas, por lo tanto:

Autoevaluación: 10

Chora López Adrián Santiago:

En la elaboración de este proyecto, pudimos reforzar los conocimientos empleando estructuras de datos estáticas y dinámicas. Siento que, en términos de complejidad, este proyecto fue menos desafiante, lo cual atribuyo a una mejor organización en la planificación de este. Esto se vio facilitado porque los esqueletos de código ya estaban estructurados y eran de constante apoyo.

Sin embargo, considero que aún hay ciertos detalles por pulir. A pesar de ello, mis compañeros de equipo me apoyaron para llevar a cabo este trabajo colaborativo de manera adecuada.

Autoevaluación: 10

Estrada Zacarías Aldo Axel:

Elaborar este proyecto me pareció muy entretenido, pude ver las mejoras que tuve respecto al inicio del curso y me hizo sentir bastante satisfecho al finalizar su elaboración. No sentí que fuera un reto tan complicado como el anterior, pero tuvo igual sus retos, ya que tuvimos que entontar la manera de hacer que fuera infinito y automático, cosa que no habíamos realizado antes, ese aspecto me gustó mucho. En el lado de la programación pude ver que no era diferente a lo que habíamos visto, por lo que era más de encontrarle la lógica a lo que hacíamos, siento que usar las estructuras de datos que vimos igual hizo que fuera más rápida su elaboración, ahorrándonos bastante tiempo al implementar varias funciones, de igual manera como equipo pudimos acoplarnos muy bien al dividirnos el trabajo.

Autoevaluación: 10

López Carreón Gabriel Augusto:

Este proyecto me sirvió para saber plantear la utilización de estructuras de datos básicas como colas lineales dinámicas, colas circulares dinámicas y pilas estáticas, lo cual permite una implementación eficiente para dar solución a una problemática dada. Asimismo, este proyecto fue más fácil de realizar gracias al proyecto anterior realizado, debido a que el bosquejo es parecido, de igual manera la manera de organización fue mucho mejor. El proyecto se divide claramente en tres fases: recepción de muestras, procesamiento y análisis, y reporte y entrega. Esto facilita la comprensión y el desarrollo por etapas del programa, por lo que fue necesario desarrollar dichas funciones en los archivos correspondientes, dependiendo de nuestras habilidades. Sin embargo, se presentaron dificultades en la gestión de las muestras y en la entrega a los departamentos solicitantes, pero trabajando colaborativamente y probando varias veces nuestros programas pudimos solucionar los errores dados para que nuestro proyecto funcione. En general tuve un gran aprendizaje, así como una retroalimentación de lo visto a lo largo del curso.

Autoevaluación: 10

VI. Rubrica de evaluación

Rúbrica para el examen-proyecto
Estructura de Datos y Algoritmos
Ing. Patricia Del Valle Morales
2º. Semestre de Ingeniería en Computación

Valor	Concepto	Excelente	Bueno	Deficiente (0%)	Puntuación
Obligatorio	1. Implementación con funciones, paso de parámetros con valor y referencia 2. Memoria dinámica 3. Apuntadores 4. Datos abstractos 5. Creación de proyecto				
20%	6. Cola lineal dinámica de datos abstractos compuestos	✓ Implementada de acuerdo con las especificaciones y sin errores ✓ Dibuja la cola de muestras de entrada correctamente cuando se inserta o eliminan elementos (20%)	✓ No se implementó con las indicaciones solicitadas ✓ Ausencia de información de las estructuras cola a pantalla (10%)	✓ Incompleto ✓ Errores en la implementación	
20%	7. Cola circular dinámica de datos abstractos compuestos	✓ Implementada de acuerdo con las especificaciones y sin errores ✓ Dibuja la cola de muestras en proceso correctamente cuando se inserta o eliminan elementos (20%)	✓ No se implementó con las indicaciones solicitadas ✓ Ausencia de información de las estructuras cola a pantalla (10%)	✓ Incompleto ✓ Errores en la implementación	
20%	8. Pila estática de datos abstractos compuestos	✓ Implementada de acuerdo con las especificaciones y sin errores ✓ Dibuja la cola de muestras por entregar correctamente cuando se inserta o eliminan elementos (20%)	✓ No se implementó con las indicaciones solicitadas ✓ Ausencia de información de las estructuras cola a pantalla (10%)	✓ Incompleto ✓ Errores en la implementación	
10%	Simulación del registro de solicitud	✓ Completo y funciona correctamente (10%)		✓ Incompleto o con errores.	
10%	Simulación del procesamiento y análisis de las muestras	✓ Completo y funciona correctamente (10%)		✓ Incompleto o con errores.	
10%	Simulación del registro de entrega y proceso de repartición a los departamentos solicitantes	✓ Completo y funciona correctamente (10%)		✓ Incompleto o con errores.	
Obligatorio	Validaciones	✓ Valida correctamente		✓ Sin implementar	
10 %	Documentación del examen-proyecto	✓ Documentación completa con reflexión		✓ Incompleta	
Obligatorio	Incluir conclusiones y reflexiones personales	(10%)			
TOTAL:					

- 1.- Los rubros marcados como obligatorios deberán de ser cubiertos al 100% de lo contrario solo obtendrá **50%** de la puntuación total alcanzada en el proyecto.
- 2.- La calificación de los integrantes del equipo será **individual** y dependerá del examen oral sobre su examen-proyecto
- 3.- El examen-proyecto será **cancelado**, colocando una calificación de **cero**, si suceden los siguientes eventos:
 - a) Si se entrega fuera de tiempo.
 - b) Se desarrolla en equipo de 3 personas, de no cumplir este requisito se cancelará.
 - c) Si existen simuladores muy parecidos
 - d) Si no cumple con los requisitos de simulación solicitados
 - e) Si no cumple con las especificaciones del examen-proyecto (sin excepción)