## Universidad Mariano Gálvez de Guatemala Facultad de Ingeniería en Sistemas de Información

Carné: 0909-20-10990 Nombre: Juan Pablo Salazar Barrios EXAMEN FINAL EN LINEA 0909 Carrera: Ingeniería en Sistemas de Información Código: Programación I 012 Asignatura: Código: **VARIANTE** Α 4 al 06/06/2021 Ciclo: 3ro. Fecha: Jornada Vespertina CAL FICACION: 100 pts. Absoluta: Ing. Amilcar Josimar Sagastume Díaz Catedrático: Relativa: 35 pts. 3 días. Semestre: Sección: Duración del examen: 10 Vo. Bo Observaciones EXAMEN: PRIMER PARCIAL: SEGUNDO PARCIAL: **EXTRAORDINARIO:** FINAL: RECUPERACIÓN: х PUNTUACION: INSTRUCCIONES: Serie I: 10 Pts. GENERALES: Resuelva lo mejor posible cada uno de los temas que se le presentan. Serie II: 10 Pts. Sea claro y ordenado, no se permite el uso de corrector, los tachones y III: 10 Pts. Serie

TEMARIO:

Serie I. (10 pts.) 1 punto c/u. Instrucciones: Responda las preguntas que se le indican a continuación, sea breve y conciso en sus respuestas. Valor de cada ítem 1 punto.

Serie IV: 5 Pts.

1. Que es la asignación de memoria dinámica, explique:

Borrones serán considerados como respuestas malas.

La memoria dinámica se está ejecutando preguntando al usuario que cantidad de memoria utilizara. Es utilizada para pedir espacio exacto que se utilizara dentro de la memoria de manera que no se desperdicie nada.

La memoria dinámica tiene 4 partes las cuales son:

- Texto
- 2. datos estáticos
- 3. pila
- 4. zona libre
- 2. Que es un apuntador en C++:

Este es utilizado cuando se hace uso de la memoria dinámica, un puntero o apuntador es una variable que se usa para apuntar a una parte de la memoria del ordenador a través de la dirección que se le da para ubicar un dato (el cual se está apuntando). Los punteros se utilizan cuando se desean implementar arrays dinámicos, árboles, tablas, listas enlazadas entre otros.

3. Qué relación hay entre apuntadores y arreglos en C++:

La relación existente se produce debido a que el término "array" es utilizado en los dos casos, ya que con el arreglo se busca entre la variable y el apuntador busca entre la memoria el dato.

Un ejemplo de sus declaraciones:

int array[7];

## int \* ptr;

- 4. Menciones algunos gestores de base de datos y cuales son más usados en C++:
  - a. Oracle Database
  - b. Microsoft Access
  - c. MvSQL
  - d. Microsoft SQL Server
- 5. Que es un árbol abb en C++, explique o grafique:

Un árbol en programación es un tipo de datos abstractos "TAD" su forma gráfica es como un árbol invertido donde al inicio esta el "tronco" y las "ramas" van abajo.

Un árbol "ABB" significa Árbol Binario de Búsqueda, Hace referencia a la forma de su estructura, pues este árbol solo tendrá dos ramas y los hijos solo podrán tener dos hijos sucesivamente. Este tipo de estructura dicta que sus elementos del lado derecho serán mayores que del lado izquierdo.

6. Que es un árbol avl en C++, explique o grafique.

Un árbol AVL hace referencia a sus inventores Adelson-Velskii y Landis, es un árbol binario de búsqueda la diferencia con el árbol ABB es que el lado izquierdo y derechos no difieren en más de uno.

Son arboles no perfectamente equilibrados, pero si lo suficientemente equilibrados para reemplazar el árbol ABB cuando este no da tiempos de búsquedas óptimos.

7. Escriba en código C++, la estructura de control DO WHILE:

Permita la misma funcionalidad que un ciclo WHILE esto quiere decir que su código se ejecuta mientras una condición se cumpla. La diferencia es que en la estructura Do While ejecuta el código antes de determinar si la condición se cumple o no.

## **EJEMPLO:**

8. Que es una lista enlazada en C++, explique o grafique:

Son estructuras de datos que permiten realizar un conjunto donde organizaremos los mismos. Está compuesta por nodos. Existen varios tipos de listas están las doblemente enlazadas, simplemente enlazada, circular simplemente enlazada y circular doblemente enlazada.

9. A que se refiere con los conceptos de Seguridad, Integridad y Concurrencia en Base de datos:

La seguridad en base de datos se refiere a la forma en la que el gestor de bases de datos protege la bd y la forma en que se evita el acceso a los datos sin autorización.

Integridad: Se refiere a una conexión verídica a la fuente en este caso a la bd, la capacidad de la base de datos para resolver la mayoría de los problemas que puede llegar a presentar los datos y su coherencia.

Concurrencia: Se tiene un control de acceso a la bd de varios usuarios a la vez y la modificación de los mismos datos en un mismo tiempo por lo que se crea algún tipo de control para que la modificación de un usuario no interfiera con otro usuario.

10. Cuál es la diferencia y el funcionamiento entre una Pila y una Cola en C++, explique:

En la pila el último dato en ingresar es el primero que se eliminará, mientras en la cola el primer dato ingresado será el primero en ser eliminado.

## LINK A GITHUB: https://github.com/PiSoftware/examen\_final\_progra1.git

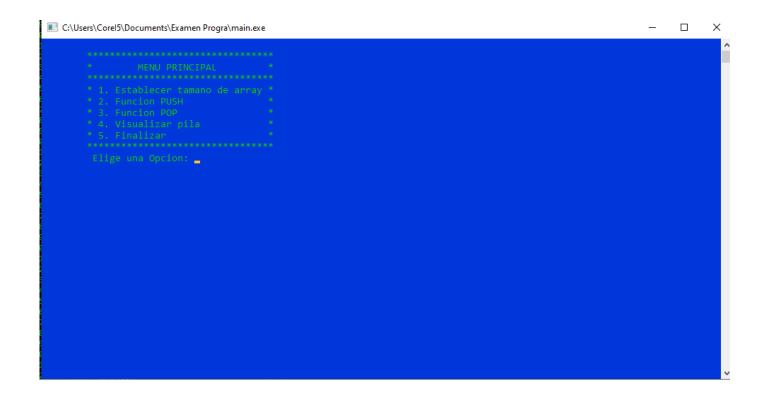
Serie II (10 Puntos) Instrucciones: Realice el ejercicio que se le propone a continuación, utilice Dev C++, el código lo puede pegar en este documento y además debe subirlo a un repositorio de GitHub, compartir el enlace.

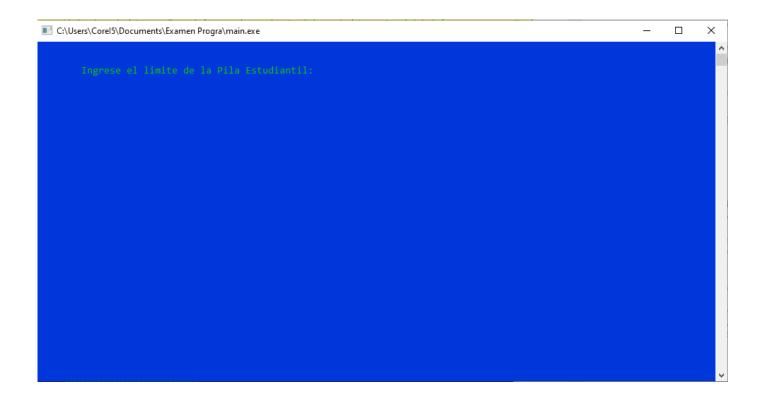
9. Realice un programa en C++, utilizando un array que almacene las notas de estudiantes respectivamente, del cual pueda aplicar las funciones Push y Pop, mostrar las opciones en un menú, para el ingreso y egreso de los datos así para la impresión de datos del array:

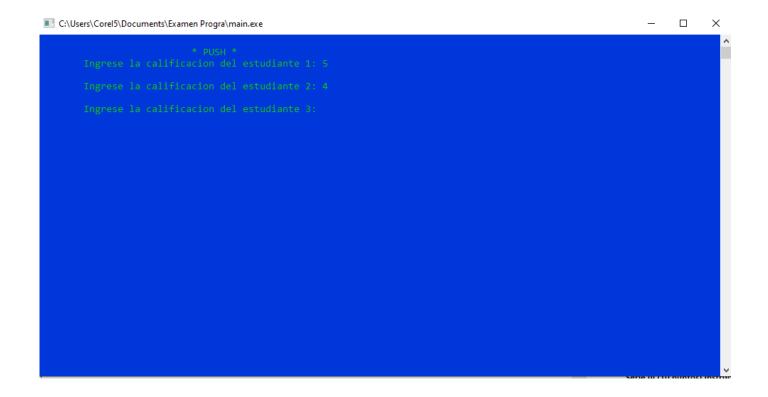
```
Presente un MENU de opciones:
Opción 1: Establecer tamaño de array.
Opción 2: Función PUSH
Opción 3: Función POP
Opción 4: Mostrar Pila.
Opción 5: Finalizar.
/*Ingenieria en Sistemas
 Juan Pablo Salazar Barrios
 Carnet: 0909-20-10990
 Tercer Semestre*/
//Declaracion de libreras
#include <iostream>
#include <conio.h>
#include <windows.h>
#include <stdlib.h>
using namespace std;
//Declaramos las variables a utilizar
int e pila;
int opcion;
int pila[1]; //inicializamos el array
//funcion para determinar el tamaño del vector
int size_pila(){
 cout<<"\n\n\tIngrese el limite de la Pila Estudiantil: ";
 cin>>e pila; //leer variable
}//fin de funcion size_pila
//Insertamos los datos
int push(){
 cout<<"\n\t
                        * METODO PUSH *"; //Titulo
```

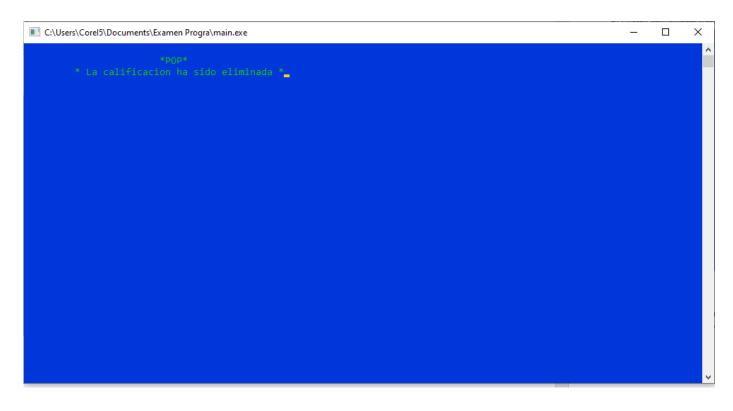
```
//for para ingresar los datos
 for (int i=0; i<e pila; i++){
  cout<<"\n\tIngrese la calificacion del estudiante " << i+1 << ": ";</pre>
  cin>>pila[i];//leemos los datos ingresados
}//fin de funcion push
//Eliminar los datos
int pop(){
 pila[e_pila-1] = 0;//inicio de pila
                     *POP*
 cout<<"\n\t
 cout <<"\n\t * La calificacion ha sido eliminada *"; //Mensaje a usuario
 _getch();//detener programa
//Visualizar los datos
int mostrar(){
                            //numero
  cout<<"\n\tCalificacion del estudiante " << e_pila << ": " << pila[e_pila-1];//datos
  e pila--; }
  while (e_pila!=0);
  cout<<"\n\n";//salto de linea
 _getch(); //detener programa
int main(){
 do{ system("color 1A"); //Color de la pantalla
   system("cls"); //limpiar la pantallla
       //visualizar menu de opciones
  cout<<"\n\t****************************
  cout << "\n\t*
                   MENU PRINCIPAL
  cout<<"\n\t**************
  cout<<"\n\t* 1. Establecer tamano de array *";
  cout<<"\n\t* 2. Funcion PUSH
  cout<<"\n\t* 3. Funcion POP
  cout<<"\n\t* 4. Visualizar pila
  cout<<"\n\t* 5. Finalizar
  cout<<"\n\t********
  cout<<"\n\t Elige una Opcion: ";
  cin>>opcion; //leer opcion
  system("cls"); //Limpiamos la pantalla
```

```
switch(opcion){
   case 1: system("cls"); //Limpiamos la pantalla
    size_pila(); //Implementacion de la funcion size_pila
    break;
   case 2:system("cls"); //Limpiamos la pantalla
    push(); //Implementacion de la funcion push
    break;
   case 3:system("cls"); //Limpiamos la pantalla
    pop(); //Implementacion de la funcion pop
    break;
   case 4:system("cls"); //Limpiamos la pantalla
    mostrar(); //Implementacion de la funcion mostrar
    break;
   case 5: break; //Evitamos que ingrese a la opcion de error
   default:{ system("color C0");//color de pantalla
                cout<<"\tERROR: El numero que ingreso es incorrecto!"; //Mensaje de error
                _getch();//detener el porgrama para leer
                break; }//mensaje de error
  }//fin default
 } while(opcion!=5);
}//fn de main
/*Ingenieria en Sistemas
 Juan Pablo Salazar Barrios
 Carnet: 0909-20-10990
 Tercer semestre*/
```





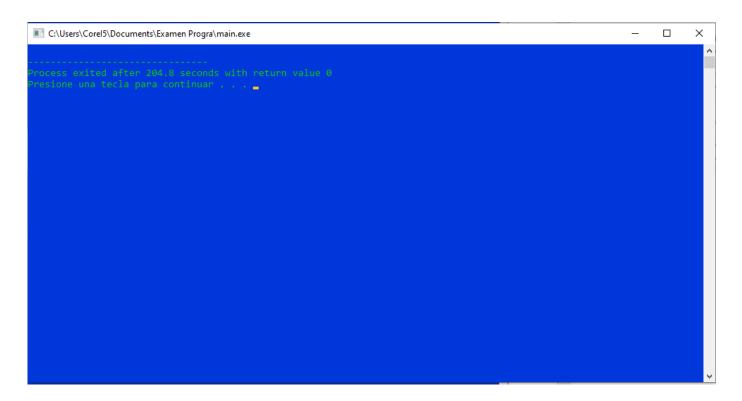




```
■ C:\Users\Corelf\Documents\Examen Progra\main.exe

- □ ×

calification del astudiante 5: 0
calification del estudiante 3: 6
Calification del estudiante 2: 4
calification del estudiante 1: 5
```



Serie III (10 puntos) Instrucciones: Realice un programa en C++, utilice Dev C++, el código lo puede pegar en este documento y además debe subirlo a un repositorio de GitHub, compartir el enlace.

10. Realice un programa en C++, utilizando un array que almacene los nombres de clientes respectivamente, del cual pueda aplicar las funciones Queue y Enqueue, mostrar las opciones en un menú, para el ingreso y egreso de los datos así para la impresión de datos del array:

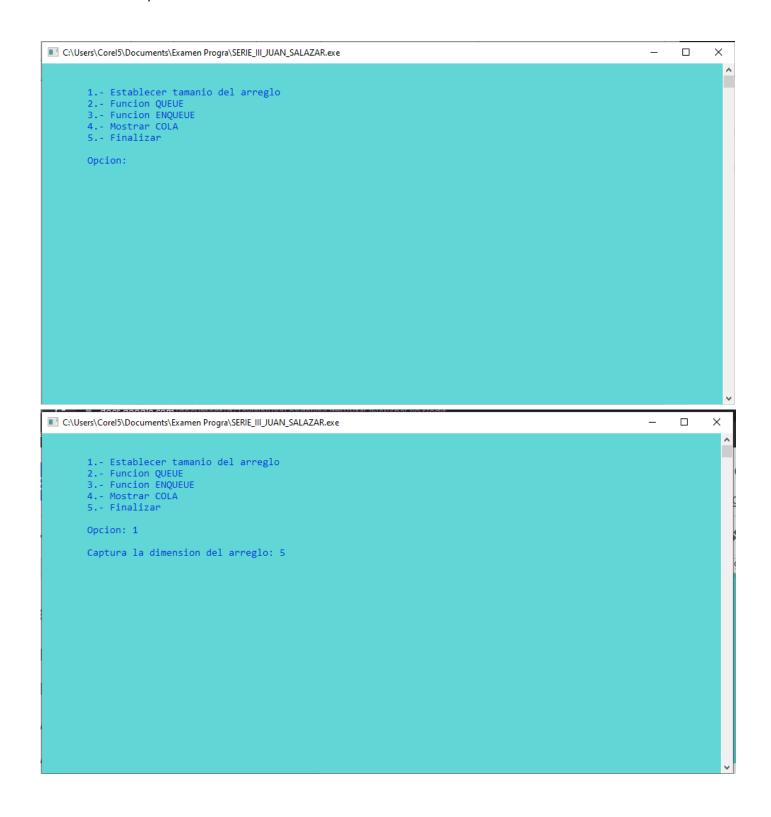
Presente un MENU de opciones:

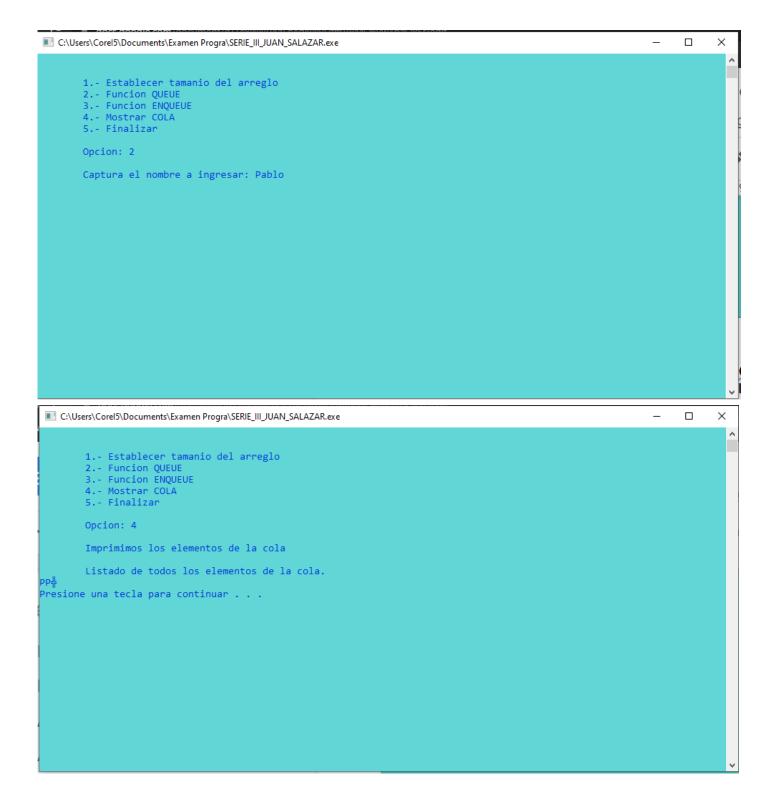
```
Opción 1: Establecer tamaño de array.
Opción 2: Función Queue
Opción 3: Función Enqueue
Opción 4: Mostrar Cola.
Opción 5: Finalizar.
/*Ingenieria en Sistemas
 Juan Pablo Salazar Barrios
 Carnet: 0909-20-10990
 Tercer Semestre*/
#include <iostream>
#include <string.h>
#include <stdlib.h>
using namespace std;
/* run this program using the console pauser or add your own getch, system("pause") or input loop */
struct Cola{
        int frente;
        int final;
        int capacidad;
        char *arreglo;
};
struct Cola* crearCola(int capacidad){
        struct Cola* cola = (struct Cola*)malloc(sizeof(struct Cola)); //Reserva del bloque de memoria
        cola->capacidad = capacidad; //Asignamos el tamaño de la cola
        cola->frente = -1; //Inicializamos el valor en el tope de la cola
        cola->final=0;//Inicialiazamos el valor del final de la cola
        cola->arreglo = (char*)malloc(cola->capacidad*sizeof(char)); //Creamos un arreglo que almacenara los datos
        return cola;
}
//FUNCION QUE UTILIZAMOS PARA REVISAR SI LA PILA ESTA VACIA
int vacia(Cola q){
        if (q.final==0)
                return (1);
return(0);
}
//FUNCION QUE DETERMINA SI LA COLA ESTA LLENA
int llena(Cola *q){
        if (q->final==q->capacidad) return (1);
 return(0);
```

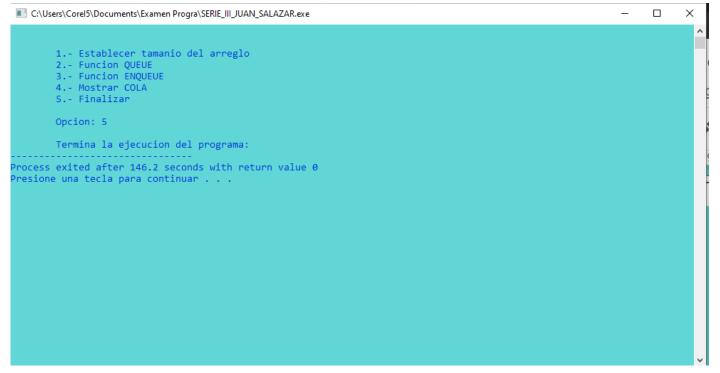
```
}
//INSERTAMOS ELEMENTOS EN LA COLA
void insertar(Cola *q, char *dato){
        q->arreglo[q->final]=*dato;
        q->final++;
}
//FUNCION QUE UTILIZAMOS PARA ELIMINAR UN ELEMENTO DE LA COLA
string eliminar(Cola *cola){
        int i;
        string x;
        x=cola->arreglo[0];
        for(i=0;i<cola->final-1;i++)
                cola->arreglo[i]=cola->arreglo[i+1];
        cola->final--;
        return(x);
}
//FUNCION PARA MOSTRAR LA INFORMACION DE LA PILA
void mostrar(struct Cola* q){
        struct Cola p=*q;
  int i;
        if(!vacia(p)){
          cout<<"\n\tListado de todos los elementos de la cola.\n";</pre>
          for(i=0;i<q->final;i++)
                cout<<q->arreglo<<endl;
        }
        else
                printf("\nLa cola se encuentra vacia.\n");
}
int main(int argc, char** argv) {
        int op;
        int tam;
        char *nombre;
```

```
struct Cola* cola;
       do{
               system("cls");
               cout<<"\n\n\t1.- Establecer tamanio del arreglo"<<endl;</pre>
               cout<<"\t2.- Funcion QUEUE"<<endl;
               cout<<"\t3.- Funcion ENQUEUE"<<endl;
               cout<<"\t4.- Mostrar COLA"<<endl;
               cout<<"\t5.- Finalizar";
               cout<<"\n\n\tOpcion: ";
               cin>>op;
               switch(op){
                       case 1: cout<<"\n\tCaptura la dimension del arreglo: ";
                                       cin>>tam;
                                       cola=crearCola(tam); //Creamos la cola
                               break;
                       case 2: cout<<"\n\tCaptura el nombre a ingresar: ";
                       nombre = (char*)malloc(sizeof(char)); //Creamos un arreglo que almacenara los datos
                                       cin>>nombre;
                                       insertar(cola, nombre);
                                       break;
                       case 3: cout<<"\n\tEliminamos un elemento de la cola";
                                       eliminar(cola);
                                       cout<<"\n\tSe ha eliminado un elemento";
                                       break;
                       case 4: cout<<"\n\tImprimimos los elementos de la cola\n";
                                       mostrar(cola);
                                       system("pause");
                                       break;
                       case 5: cout<<"\n\tTermina la ejecucion del programa:";
                                       break;
                       default: cout << "\n\tOpcion no valida";
                                       break;
               }
       }while(op!=5);
       return 0;
/*Ingenieria en Sistemas
Juan Pablo Salazar Barrios
Carnet: 0909-20-10990
```

}







Serie IV (5 puntos) Instrucciones: Realice un programa en C++, utilice Dev C++, el código lo puede pegar en este documento y además debe subirlo a un repositorio de GitHub, compartir el enlace.

10. Realice un programa en C++, utilizando un árbol abb que almacene números respectivamente, del cual al insertar un dato se almacene en los nodos, mostrar las opciones en un menú, para el ingreso, borrado y búsqueda de los datos entre los nodos, así como para la impresión del árbol:

Presente un MENU de opciones:

using namespace std;

struct nodo \*izq, \*der;

struct nodo{
 int nro;

**}**;

Opción 1: Insertar un nodo en el árbol.
Opción 2: Mostrar el árbol completo
Opción 3: buscar nodo especifico
Opción 4: recorrer el árbol, pre-orden, in-orden, pos-orden
Opción 5: borrar un nodo del árbol
Opción 6: finalizar.

/\*Ingenieria en Sistemas
Juan Pablo Salazar Barrios
Carnet: 0909-20-10990
Tercer Semestre\*/

#include <iostream>
#include <cstdlib>

```
typedef struct nodo *ABB;
/* es un puntero de tipo nodo que hemos llamado ABB, que ulitizaremos
 para mayor facilidad de creacion de variables */
ABB crearNodo(int x)
  ABB nuevoNodo = new(struct nodo);
  nuevoNodo->nro = x;
   nuevoNodo->izq = NULL;
  nuevoNodo->der = NULL;
  return nuevoNodo;
}
void insertar(ABB & arbol, int x)
  if(arbol==NULL)
      arbol = crearNodo(x);
  else if(x < arbol->nro)
     insertar(arbol->izq, x);
  else if(x > arbol->nro)
     insertar(arbol->der, x);
}
void preOrden(ABB arbol)
  if(arbol!=NULL)
     cout << arbol->nro <<" ";
     preOrden(arbol->izq);
     preOrden(arbol->der);
}
void enOrden(ABB arbol)
  if(arbol!=NULL)
     enOrden(arbol->izq);
     cout << arbol->nro << " ";
     enOrden(arbol->der);
  }
}
void postOrden(ABB arbol)
  if(arbol!=NULL)
     postOrden(arbol->izq);
```

```
postOrden(arbol->der);
     cout << arbol->nro << " ";
  }
}
void verArbol(ABB arbol, int n)
  if(arbol==NULL)
     return;
  verArbol(arbol->der, n+1);
  for(int i=0; i<n; i++)
     cout<<" ";
   cout<< arbol->nro <<endl;
   verArbol(arbol->izq, n+1);
}
bool buscar(ABB arbol, int dato)
   int r=0; // 0 indica que no lo encontre
  if(arbol==NULL)
    return r;
  if(dato<arbol->nro)
     r = buscar(arbol->izq, dato);
   else if(dato> arbol->nro)
     r = buscar(arbol->der, dato);
   else
        r = 1; // son iguales, lo encontre
   return r;
}
int main()
  ABB arbol = NULL; // creado Arbol
  system("color A1"); //Color de la pantalla
  int n; // numero de nodos del arbol
  int x; // elemento a insertar en cada nodo
        int op;
        int num;
  cout << "\n\t\t ARBOL BINARIO DE BUSQUEDA \n\n";</pre>
```

```
cout<<"1.- Insertar un nodo en el arbol"<<endl;
cout<<"2.- Mostrar el arbol completo"<<endl;
cout<<"3.- Buscar un nodo especifico"<<endl;
cout<<"4.- Recorrer el arbol, pre-orden, in-orden, pos-orden"<<endl;
cout<<"5.- Borrar un nodo del arbol"<<endl;
cout<<"6.- Finaliza"<<endl;
cout<<"Opcion: ";
cin>>op;
switch(op){
        case 1: cout << " Captura el numero de nodos del arbol: ";
                cin >> n;
                cout << endl;
                for(int i=0; i<n; i++)
                cout << " Numero del nodo " << i+1 << ": ";
                cin >> x;
                insertar( arbol, x);
                }
                break;
       case 2: cout << "\n Mostrando ABB \n\n";
                verArbol( arbol, 0);
                break;
case 3: cout<<"Captura el nodo a buscar: ";
                cin>>num;
                if(buscar(arbol, num)){
                                cout<<"Nodo encontrado"<<endl;
                                getch();
                        }else{
                                cout<<"Nodo no encontrado"<<endl;
                                getch();
                        }
       case 4: cout << "\n Recorridos del ABB";
                        cout << "\n\n Pre Orden : "; preOrden(arbol);</pre>
                  cout << "\n\n In orden : "; enOrden(arbol);</pre>
                  cout << "\n\n Pos Orden : "; postOrden(arbol);</pre>
        case 5: cout<<"Introduzca el nodo a eliminar";
                        //Funcion Elimina Nodo
                        break;
       case 6: cout<<"Finaliza el programa";
```

break;

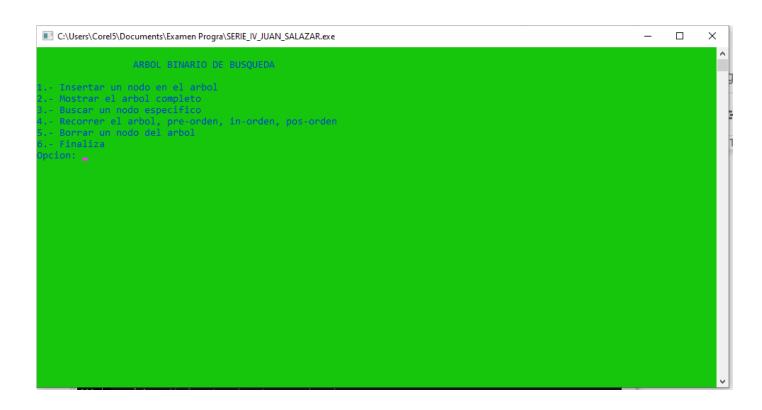
```
}
}

}while(op!=6);

cout << endl << endl;

system("pause");
 return 0;
}

/*Ingenieria en Sistemas
Juan Pablo Salazar Barrios
Carnet: 0909-20-10990
Tercer Semestre*/</pre>
```



```
×
C:\Users\Corel5\Documents\Examen Progra\SERIE_IV_JUAN_SALAZAR.exe
                   ARBOL BINARIO DE BUSQUEDA
   Insertar un nodo en el arbol
 - Mostrar el arbol completo
  - Buscar un nodo especifico
 - Recorrer el arbol, pre-orden, in-orden, pos-orden
- Borrar un nodo del arbol
.- Finaliza
pcion: 1
Captura el numero de nodos del arbol: 5
Numero del nodo 1: 2
Numero del nodo 2: 4
Numero del nodo 3: 6
Numero del nodo 4: 8
                                                                                                                                       ■ C:\Users\Corel5\Documents\Examen Progra\SERIE_IV_JUAN_SALAZAR.exe
aptura el nodo a buscar: 3
lodo no encontrado
Recorridos del ABB
Pre Orden : 2 4 6 8 9
In orden : 2 4 6 8 9
Pos Orden : 9 8 6 4 2 Introduzca el nodo a eliminar1.- Insertar un nodo en el arbol
 .- Mostrar el arbol completo
- Buscar un nodo específico
- Recorrer el arbol, pre-orden, in-orden, pos-orden
- Borrar un nodo del arbol
 .- Finaliza
```

aptura el nodo a buscar: 4

lodo encontrado

