

INFORME CAPTURAS

REPOSITORIO DATOS AVANZADA

LINK: <https://github.com/Esteban1426/Repositorio-Datos-Avanzada>

TRABAJO 3: Programas de listas Simple, Circular y Doblemente enlazada

1. PRIMER PROGRAMA LISTA SIMPLE:

```
#include <iostream>

using namespace std;

struct nodo
{
    int num;
    nodo*siguiente;
};

nodo * cabeza = nullptr;
nodo * cola = nullptr;

void Insertar(int valor)
{
    nodo * nuevo = new nodo();
    nuevo -> num = valor;

    if (cabeza == nullptr)
    {
        cabeza = nuevo;
        cola = nuevo;
    }
    else
    {
        cola -> siguiente = nuevo;
        cola = nuevo;
    }

    nodo * actual = cabeza;
    while (actual != nullptr)
    {
```

```

        while (actual != nullptr)
        {
            actual = actual -> siguiente;
        }
    }

    void mostrar()
    {
        int contador = 1;

        if(!cabeza)
        {
            cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
            return;
        }

        nodo * actual = cabeza;
        do
        {
            cout << "Nodo numero "<< contador << " : " << actual->num << " " << endl;
            actual = actual -> siguiente;
            contador++;
        } while (actual != nullptr);
    }

```

```

    void Busqueda(int valor)
    {
        int posicion = 1;
        bool encontrado = false;

        if(!cabeza)

```

```

        {
            cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
            return;
        }

        nodo * actual = cabeza;

        do
        {
            if(actual -> num == valor)
            {
                cout<<"El numero "<< valor <<" se encuentra en el nodo "<< posicion << " y su direccion de memoria es: "<< actual << endl;
                encontrado = true;
            }

            actual = actual -> siguiente;
            posicion ++;
        }while(actual != nullptr);

        if (lencontrado)
        {
            cout << "Numero no encontrado" << endl;
        }
    }

```

```

    void Actualizar (int posicion, int nuevoValor)
    {
        if (!cabeza)
        {

```

```

void Actualizar (int posicion, int nuevoValor)
{
    if (!cabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
        return;
    }

    nodo * actual = cabeza;
    int contador = 1;

    do
    {
        if (contador == posicion)
        {
            actual -> num = nuevoValor;
            cout << "El valor del nodo numero "<< posicion << " se actualizo a este valor: "<< nuevoValor << endl;
            return;
        }
        actual = actual -> siguiente;
        contador++;
    }while (actual != nullptr);

    cout << "Posicion invalida" << endl;
}

void Borrar(int posicion)
{
    if (!cabeza)
void Borrar(int posicion)
{
    if (!cabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
        return;
    }

    nodo * borrar = nullptr;

    if (posicion == 1)
    {
        borrar = cabeza;
        cabeza = cabeza -> siguiente;
        delete borrar;
        cout << "El nodo de la posicion "<< posicion << " Ha sido borrado correctamente" << endl;
        return;
    }

    nodo * anterior = nullptr;
    nodo * actual = cabeza;

    int contador = 1;

    do
    {
        if (contador == posicion)
        {
            borrar = actual;
            anterior -> siguiente = actual -> siguiente;
            delete borrar;
            cout << "El nodo de la posicion "<< posicion << " Ha sido borrado correctamente" << endl;

```

```

        cout << "El nodo de la posicion "<< posicion << " Ha sido borrado correctamente" << endl;
        return;
    }
    anterior = actual;
    actual = actual-> siguiente;
    contador++;
}while(actual != nullptr);

cout << "La posicion ingresada es invalida" << endl;
}

int main()
{
    int opcion;
    char continuar = 'y';
    do
    {
        cout << " Ingrese el numero correspondiente a la opcion que desees: "<<endl;
        cout << " 1. Ingresar valores a los nodos "<<endl << " 2. Busqueda de valores " <<endl << " 3. Actualizar el valor de un nodo "<< endl;
        cout << " Respuesta: ";
        cin >> opcion;

        switch (opcion)
        {
            case 1:
            {
                int valor;

                do
                {
                    cout << "Ingrese un numero /*Nota: Si quieres salir ingresa numero 0*/ : "<< endl;

                    {
                        cout << "Ingrese un numero /*Nota: Si quieres salir ingresa numero 0*/ : "<< endl;
                        cin >> valor;

                        if(valor != 0)
                        {
                            Insertar(valor);
                        }
                    }

                } while (valor != 0);

                cout << "Lista Simple: " << endl;
                mostrar();

                cout << "Deseas Continuar con el programa( y / n ) en minusculas: " <<endl;
                cin >>continuar;

                break;
            }

            case 2:
            {
                int numeroBuscar;
                cout <<"Digite el valor que desea buscar: " << endl << "Respuesta: ";
                cin >> numeroBuscar;

                Busqueda(numeroBuscar);

                cout << "Deseas Continuar con el programa( y / n ) en minusculas: " <<endl;
                cin >>continuar;

                break;
            }
        }
    }
}

```

```
case 3:
{
    int posicion;
    int nuevoValor;

    cout << "Digita la posicion del nodo en la que quieres cambiar el valor: " << endl << "Respuesta: ";
    cin>> posicion;

    cout << "Digita el valor nuevo para el nodo: " << endl << "Respuesta: ";
    cin>> nuevoValor;

    Actualizar(posicion, nuevoValor);

    cout << "Deseas Continuar con el programa( y / n ) en minusculas: " <<endl;
    cin >>continuar;
    break;
}

case 4:
{
    int posicion;
    cout << " Ingresa la posicion del nodo que quieres eliminar: " << endl << "Respuesta: ";
    cin >> posicion;

    Borrar(posicion);
    cout << "Lista Simple Actual: " <<endl;
    mostrar();

    cout << "Deseas Continuar con el programa( y / n ) en minusculas: " <<endl;
    cin >>continuar;

    break;
```

```

        case 5:
        {
            cout << " Esta es la lista Actual: " << endl;
            mostrar();

            cout << "Deseas Continuar con el programa( y / n ) en minusculas: " <<endl;
            cin >>continuar;

            break;
        }

        case 6:
        {
            cout << " Vale, Saliendo... " << endl;
            break;
        }

        default:
        {
            cout <<"Opcion no valida";
            break;
        }
    }

}while(continuar != 'n' && opcion != 6);

nodo * actual = cabeza;
while (actual != nullptr)
{
    nodo * siguiente = actual -> siguiente;
    delete actual;
    actual = siguiente;
}

```

EJECUCION CMD:

- **Primera Opción:**

```
Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 1
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
1
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
2
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
3
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
4
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
0
Lista Simple:
Nodo numero 1 :1
Nodo numero 2 :2
Nodo numero 3 :3
Nodo numero 4 :4
Deseas Continuar con el programa( y / n ) en minusculas:
_
```

- **Segunda Opción:**

```
Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 2
Digite el valor que desea buscar:
Respuesta: 3
El numero 3 se encuentra en el nodo 3 y su direccion de memoria es: 0x2b44bca17d0
Deseas Continuar con el programa( y / n ) en minusculas:
```

- **Tercera Opción:**


```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 3
Digita la posicita del nodo en la que quieres cambiar el valor:
Respuesta: 2
Digita el valor nuevo para el nodo:
Respuesta: 3
El valor del nodo numero 2 se actualizo a este valor: 3
Deseas Continuar con el programa( y / n ) en minusculas:

```

- **Cuarta Opción:**

```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 4
Ingresa la posiccion del nodo que quieres eliminar:
Respuesta: 4
El nodo de la posiccion 4 Ha sido borrado correctamente
Lista Actual:
Nodo numero 1 :1
Nodo numero 2 :3
Nodo numero 3 :3
Deseas Continuar con el programa( y / n ) en minusculas:

```

- **Quinta Opción:**

```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 5
Esta es la lista Actual:
Nodo numero 1 :1
Nodo numero 2 :3
Nodo numero 3 :3
Deseas Continuar con el programa( y / n ) en minusculas:

```

2. SEGUNDO PROGRAMA LISTA CIRCULAR:

```
#include <iostream>

using namespace std;

struct nodo
{
    int num;
    nodo*siguiente;
};

nodo*cabeza = nullptr;
nodo*cola = nullptr;

void Insertar(int valor)
{
    nodo * nuevo = new nodo();
    nuevo -> num = valor;

    if (cabeza == nullptr)
    {
        cabeza = nuevo;
        cola = nuevo;
    }
    else
    {
        cola -> siguiente = nuevo;
        nuevo -> siguiente = cabeza;
        cola = nuevo;
    }
}

void Busqueda(int valor)
```

```

void Busqueda(int valor)
{
    int posicion = 1;
    bool encontrado = false;

    if(!cabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemno valores" << endl;
        return;
    }

    nodo * actual = cabeza;

    do
    {
        if(actual -> num == valor)
        {
            cout<<"El numero "<< valor <<" se encuentra en el nodo "<< posicion << " y su direccion de memoria es: "<< actual << endl;
            encontrado = true;
        }

        actual = actual -> siguiente;
        posicion ++;

    }while(actual != cabeza);

    if (!encontrado)
    {
        cout << "Numero no encontrado" << endl;
    }
}

void Actualizar (int posicion, int nuevoValor)
{
    if (!cabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemno valores" << endl;
        return;
    }

    nodo * actual = cabeza;
    int contador = 1;

    do
    {
        if (contador == posicion)
        {
            actual -> num = nuevoValor;
            cout << "El valor del nodo numero "<< posicion << " se actualizo a este valor: "<< nuevoValor << endl;
            return;
        }
        actual = actual -> siguiente;
        contador++;

    }while (actual != cabeza);

    cout << "Posicion invalida" << endl;
}

void Borrar(int posicion)
{
    if(!cabeza)
    {

```

```
cout << "Posicion invalida" << endl;
```

```
}
```

```
void Borrar(int posicion)
```

```
{
```

```
    if(!cabeza)
```

```
    {
```

```
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
```

```
        return;
```

```
    }
```

```
    nodo * borrar = nullptr;
```

```
    if (posicion == 1)
```

```
    {
```

```
        borrar = cabeza;
```

```
        if (cabeza == cola)
```

```
        {
```

```
            cabeza = nullptr;
```

```
            cola = nullptr;
```

```
        }
```

```
        else
```

```
        {
```

```
            cabeza = cabeza -> siguiente;
```

```
            cola -> siguiente = cabeza;
```

```
        }
```

```
        delete borrar;
```

```
        cout << "El nodo de la posicion "<< posicion << " Ha sido borrado correctamente" << endl;
```

```
        return;
```

```
    }
```

```
    cout << "El nodo de la posicion "<< posicion << " Ha sido borrado correctamente" << endl;
```

```
    return;
```

```
}
```

```
nodo * anterior = nullptr;
```

```
nodo * actual = cabeza;
```

```
int contador = 1;
```

```
do
```

```
{
```

```
    if (contador == posicion)
```

```
    {
```

```
        borrar = actual;
```

```
        if (anterior)
```

```
        {
```

```
            anterior -> siguiente = actual -> siguiente;
```

```
            if (borrar == cabeza)
```

```
            {
```

```
                cabeza = actual -> siguiente;
```

```
            }
```

```
            if (borrar == cola)
```

```
            {
```

```
                cola = anterior;
```

```
            }
```

```
        }
```

```
        else
```

```
        {
```

```
            cabeza = actual -> siguiente;
```

```
        }
```

```
        delete borrar;
```

```

        cout << "El nodo de la posicion "<< posicion << " Ha sido borrado correctamente" << endl;
        return;
    }
    anterior = actual;
    actual = actual-> siguiente;
    contador++;
}while(actual != cabeza);

cout << "La posicion ingresada es invalida" << endl;
}

void mostrar()
{
    int contador = 1;
    if(!cabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
        return;
    }

    nodo * actual= cabeza;
    do
    {
        cout << "Nodo numero "<< contador << " : " <<actual->num << " " << endl;
        actual = actual -> siguiente;
        contador++;
    } while (actual != cabeza);

    cout << "Valor del Nodo cabeza: " << actual->num << " " << endl;
}

int main()
{
    int opcion;
    char continuar = 'y';
    do
    {
        cout << " Ingresa el numero correspondiente a la opcion que desees: "<<endl;
        cout << " 1. Ingresar valores a los nodos "<<endl << " 2. Busqueda de valores " <<endl << " 3. Actualizar el valor de un nodo "<<endl;
        cout << " Respuesta: ";
        cin >> opcion;

        switch (opcion)
        {
            case 1:
            {
                int valor;

                do
                {
                    cout << "Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ : "<< endl;
                    cin >> valor;

                    if(valor != 0)
                    {
                        Insertar(valor);
                    }
                } while (valor != 0);

                cout << "Lista circular Actual: " << endl;
                mostrar();
            }
        }
    } while (continuar == 'y');
}

```

```

        cout << "Deseas Continuar con el programa( y / n ) en minusculas: " <<endl;
        cin >>continuar;

        break;
    }

    case 2:
    {
        int numeroBuscar;
        cout <<"Digite el valor que desea buscar: " << endl << "Respuesta: ";
        cin >> numeroBuscar;

        Busqueda(numeroBuscar);

        cout << "Deseas Continuar con el programa( y / n ) en minusculas: " <<endl;
        cin >>continuar;

        break;
    }

    case 3:
    {
        int posicion;
        int nuevoValor;

        cout << "Digita la posicion del nodo en la que quieres cambiar el valor: " << endl << "Respuesta: ";
        cin>> posicion;

        cout << "Digita el valor nuevo para el nodo: " << endl << "Respuesta: ";
        cin>> nuevoValor;

        Actualizar(posicion, nuevoValor);
    }

```

```
case 4:
{
    int posicion;
    cout << " Ingresa la posicion del nodo que quieres eliminar: " << endl << "Respuesta: ";
    cin >> posicion;

    Borrar(posicion);
    cout << "Lista Actual: " << endl;
    mostrar();

    cout << "Deseas Continuar con el programa( y / n ) en minusculas: " << endl;
    cin >> continuar;

    break;
}

case 5:
{
    cout << " Esta es la lista Actual: " << endl;
    mostrar();

    cout << "Deseas Continuar con el programa( y / n ) en minusculas: " << endl;
    cin >> continuar;

    break;
}

case 6:
{
    cout << " Vale, Saliendo... " << endl;
    break;
}
```

```

        case 6:
        {
            cout << " Vale, Saliendo... " << endl;
            break;
        }

        default:
        {
            cout <<"Opcion no valida";
            break;
        }
    }
}while(continuar != 'n' && opcion != 6);

nodo * actual = cabeza;
while (actual != nullptr)
{
    nodo * siguiente = actual -> siguiente;
    delete actual;
    actual = siguiente;
}

cout << " Fin..";
return 0;
}

```

EJECUCION CMD:

- Primera Opción:


```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 1
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
1
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
2
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
3
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
4
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
0
Lista circular:
Nodo numero 1 :1
Nodo numero 2 :2
Nodo numero 3 :3
Nodo numero 4 :4
Nodo cabeza: 1
Deseas Continuar con el programa( y / n ) en minusculas:

```

- **Segunda Opción:**

```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 2
Digite el valor que desea buscar:
Respuesta: 3
El numero 3 se encuentra en el nodo 3 y su direccion de memoria es: 0x29019e217d0
Deseas Continuar con el programa( y / n ) en minusculas:

```

- **Tercera Opción:**

```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 3
Digita la posicion del nodo en la que quieres cambiar el valor:
Respuesta: 1
Digita el valor nuevo para el nodo:
Respuesta: 6
El valor del nodo numero 1 se actualizo a este valor: 6
Deseas Continuar con el programa( y / n ) en minusculas:

```

- **Cuarta Opción:**

```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 4
Ingresa la posicion del nodo que quieres eliminar:
Respuesta: 4
El nodo de la posicion 4 Ha sido borrado correctamente
Lista Actual:
Nodo numero 1 :6
Nodo numero 2 :2
Nodo numero 3 :3
Nodo cabeza: 6
Deseas Continuar con el programa( y / n ) en minusculas:
y

```

- **Quinta Opción:**

```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual
6. Salir del programa
Respuesta: 5
Esta es la lista Actual:
Nodo numero 1 :6
Nodo numero 2 :2
Nodo numero 3 :3
Nodo cabeza: 6
Deseas Continuar con el programa( y / n ) en minusculas:

```

3. TERCER PROGRAMA LISTA DOBLEMENTE ENLAZADA:

```
#include <iostream>

using namespace std;

int numeroNodos;

struct nodo
{
    int num;
    nodo*izq;
    nodo*dere;
};

nodo * cabeza = nullptr;
nodo * cola = nullptr;

void Insertar(int valor)
{
    nodo * nuevo = new nodo();
    nuevo -> num = valor;
    nuevo -> izq = nullptr;
    nuevo -> dere = nullptr;

    if (cabeza == nullptr)
    {
        cabeza = nuevo;
        cola = nuevo;
    }
    else
    {
        nuevo -> izq = cola;
        cola -> dere = nuevo;
    }
}
```

```

    }
    numeroNodos++;
}

void Busqueda(int valor)
{
    int posicion = 1;
    bool encontrado = false;

    if(!lcabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
        return;
    }

    nodo * actual = cabeza;

    do
    {
        if(actual -> num == valor)
        {
            cout << "El numero " << valor << " se encuentra en el nodo " << posicion << " y su direccion de memoria es: " << actual << endl;
            encontrado = true;
        }

        actual = actual -> dere;
        posicion ++;
    } while (actual != nullptr);

    if (!encontrado)
    {
        cout << "Numero no encontrado" << endl;
    }
}

void Actualizar (int posicion, int nuevoValor)
{
    if (!lcabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
        return;
    }

    nodo * actual = cabeza;
    int contador = 1;

    do
    {
        if (contador == posicion)
        {
            actual -> num = nuevoValor;
            cout << "El valor del nodo numero " << posicion << " se actualizo a este valor: " << nuevoValor << endl;
            return;
        }
        actual = actual -> dere;
        contador++;
    } while (actual != nullptr);
}

```

```

void Borrar(int posicion)
{
    if(!cabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
        return;
    }

    if (posicion == 1)
    {
        nodo * borrar = cabeza;
        cabeza = cabeza -> dere;
        if (cabeza)
        {
            cabeza -> izq = nullptr;
        }
        delete borrar;
        numeroNodos --;
        cout << "El nodo de la posicion "<< posicion << " Ha sido borrado correctamente" << endl;
        return;
    }

    nodo * actual = cabeza;

    int contador = 1;

    do
    {
        if (contador == posicion)
        {
            nodo * borrar = actual;
            if (actual -> dere)

```

```

        nodo * borrar = actual;
        if (actual -> dere)
        {
            actual -> dere -> izq = actual -> izq;
        }

        if (actual -> izq)
        {
            actual -> izq -> dere = actual -> dere;
        }
        if (actual -> dere == nullptr)
        {
            cola = actual -> izq;
        }

        delete borrar;
        numeroNodos--;
        cout << "El nodo de la posicion " << posicion << " Ha sido borrado correctamente" << endl;
        return;
    }
    actual = actual-> dere;
    contador++;
}while(actual != nullptr);

cout << "La posicion ingresada es invalida" << endl;
}

void mostrar()
{
    int contador = 1;
    if(!cabeza)

```

```

void mostrar()
{
    int contador = 1;
    if(!cabeza)
    {
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
        return;
    }

    nodo * actual= cabeza;
    do
    {
        cout << "Nodo numero "<< contador << " :" <<actual->num << " "<< endl;
        actual = actual -> dere;
        contador++;
    } while (actual != nullptr);
}

void mostrarInversa()
{
    int contador = numeroNodos;
    if(!cola)
    {
        cout << "La lista esta vacia tienes que ingresar priemro valores" << endl;
        return;
    }

    nodo * actual= cola;
    do
    {
        cout << "Nodo numero "<< contador << " :" <<actual->num << " "<< endl;

        cout << "Nodo numero "<< contador << " :" <<actual->num << " "<< endl;
        actual = actual -> izq;
        contador--;
    } while (actual != nullptr);
}

int main()
{
    int opcion;
    char continuar = 'y';
    do
    {
        cout << " Ingresa el numero correspondiente a la opcion que desees: "<<endl;
        cout << " 1. Ingresar valores a los nodos "<<endl << " 2. Busqueda de valores " <<endl << " 3. Actualizar el valor de un nodo "<<endl;
        cout << " Respuesta: ";
        cin >> opcion;

        switch (opcion)
        {
            case 1:
            {
                int valor;

                do
                {
                    cout << "Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ : "<< endl;
                    cin >> valor;

                    if(valor != 0)
                    {
                        Insertar(valor);
                    }
                }
            }
        }
    } while (continuar == 'y');
}

```

```

    } while (valor != 0);

    cout << "Lista Doblemente Enlazada Actual: " << endl;
    mostrar();
    cout << "-----" << endl;
    cout << "Lista Doblemente Enlazada Inversa Actual: " << endl;
    mostrarInversa();

    cout << "Deseas Continuar con el programa( y / n ) en minusculas: " << endl;
    cin >> continuar;

    break;
}

case 2:
{
    int numeroBuscar;
    cout << "Digite el valor que desea buscar: " << endl << "Respuesta: ";
    cin >> numeroBuscar;

    Busqueda(numeroBuscar);

    cout << "Deseas Continuar con el programa( y / n ) en minusculas: " << endl;
    cin >> continuar;

    break;
}

case 3:
{
    int posicion;
    int nuevoValor;

case 3:
{
    int posicion;
    int nuevoValor;

    cout << "Digita la posicion del nodo en la que quieres cambiar el valor: " << endl << "Respuesta: ";
    cin >> posicion;

    cout << "Digita el valor nuevo para el nodo: " << endl << "Respuesta: ";
    cin >> nuevoValor;

    Actualizar(posicion, nuevoValor);

    cout << "Deseas Continuar con el programa( y / n ) en minusculas: " << endl;
    cin >> continuar;
    break;
}

case 4:
{
    int posicion;
    cout << " Ingresa la posicion del nodo que quieres eliminar: " << endl << "Respuesta: ";
    cin >> posicion;

    Borrar(posicion);
    cout << "Lista Actual: " << endl;
    mostrar();
    cout << "-----" << endl;
    cout << "Esta es la lista Actual inversa" << endl;
    mostrarInversa();

```



```

    case 5:
    {
        cout << "Esta es la lista Actual: " << endl;
        mostrar();
        cout << "-----" << endl;
        cout << "Esta es la lista Actual inversa" << endl;
        mostrarInversa();

        cout << "Deseas Continuar con el programa( y / n ) en minusculas: " << endl;
        cin >>continuar;

        break;
    }

    case 6:
    {
        cout << " Vale, Saliendo... " << endl;
        break;
    }

    default:
    {
        cout <<"Opcion no valida";
        break;
    }
}

while(continuar != 'n' && opcion != 6);

odo * actual = cabeza;
while (actual != nullptr)

```

EJECUCION CMD:

- **Primer Opción:**

```

Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual y Mostrar la lista actual inversa
6. Salir del programa
Respuesta: 1
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
1
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
2
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
3
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
4
Ingresa un numero /*Nota: Si quieres salir ingresa numero 0*/ :
0
Lista Doblemente Enlazada:
Nodo numero 1 :1
Nodo numero 2 :2
Nodo numero 3 :3
Nodo numero 4 :4
-----

Lista Doblemente Enlazada Inversa:
Nodo numero 4 :4
Nodo numero 3 :3
Nodo numero 2 :2
Nodo numero 1 :1
Deseas Continuar con el programa( y / n ) en minusculas:

```

- **Segunda Opción:**

```

Lista Doblemente Enlazada Inversa:
Nodo numero 4 :4
Nodo numero 3 :3
Nodo numero 2 :2
Nodo numero 1 :1
Deseas Continuar con el programa( y / n ) en minusculas:
y
Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual y Mostrar la lista actual inversa
6. Salir del programa
Respuesta: 2
Digite el valor que desea buscar:
Respuesta: 4
El numero 4 se encuentra en el nodo 4 y su direccion de memoria es: 0x1f7b23b17f0
Deseas Continuar con el programa( y / n ) en minusculas:

```

- **Tercera Opción:**

```
Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual y Mostrar la lista actual inversa
6. Salir del programa
Respuesta: 3
Digita la posicion del nodo en la que quieres cambiar el valor:
Respuesta: 2
Digita el valor nuevo para el nodo:
Respuesta: 4
El valor del nodo numero 2 se actualizo a este valor: 4
Deseas Continuar con el programa( y / n ) en minusculas:
```

- **Cuarta Opción:**

```
Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual y Mostrar la lista actual inversa
6. Salir del programa
Respuesta: 4
Ingresa la posicion del nodo que quieres eliminar:
Respuesta: 4
El nodo de la posicion 4 Ha sido borrado correctamente
Lista Actual:
Nodo numero 1 :1
Nodo numero 2 :4
Nodo numero 3 :3
-----

Esta es la lista Actual inversa
Nodo numero 3 :3
Nodo numero 2 :4
Nodo numero 1 :1
Deseas Continuar con el programa( y / n ) en minusculas:
```

- **Quinta Opción:**

```
Ingresa el numero correspondiente a la opcion que desees:
1. Ingresar valores a los nodos
2. Busqueda de valores
3. Actualizar el valor de un nodo
4. Borrar un nodo
5. Mostrar la lista actual y Mostrar la lista actual inversa
6. Salir del programa
Respuesta: 5
Esta es la lista Actual:
Nodo numero 1 :1
Nodo numero 2 :4
Nodo numero 3 :3
-----

Esta es la lista Actual inversa
Nodo numero 3 :3
Nodo numero 2 :4
Nodo numero 1 :1
Deseas Continuar con el programa( y / n ) en minusculas:
```