

Cola Circular y cola dinámica

1. Ejecute el programa **Dev-C++**.
2. En el menú **Archivo** seleccione la opción **Nuevo** y a continuación **Archivo Fuente**.
3. En el nuevo documento transcriba el código del siguiente programa, en el cual se utiliza un arreglo para modelar una cola circular, controlándola mediante dos índices: Frente y Final, para indicar la ubicación de los elementos del inicio y del final de la cola dentro del arreglo y evitar que se traslapen.
4. En la cola se realizan las operaciones de **Put** y **Get**, para agregar o quitar elementos de la cola, el inconveniente principal de este modelo es que la cola no puede rebasar el tamaño máximo fijado, que para este ejercicio es de solo **5** elementos:

```
#include<iostream>
using namespace std;

const int MAXCOLA=5;

template<class T>
class ColaCircular {
    private: T EspacioCola[MAXCOLA];
            int Frente,Final;

    public: ColaCircular();
            int Put(const T);
            int Get();
            int ColaLlena()const;
            int ColaVacía()const;
            void VerCola(void)const;
};

template<class T>
ColaCircular<T>::ColaCircular() {
    Frente=MAXCOLA-1;
    Final=MAXCOLA-1;
}

template<class T>
int ColaCircular<T>::ColaVacía(void) const {
    if (Frente==Final) return 1;
    else return 0;
}
```

```
template<class T>
int ColaCircular<T>::ColaLlena(void) const {
    if((Final+1)%MAXCOLA==Frente) return 1;
    else return 0;
}

template<class T>
int ColaCircular<T>::Put(const T Dato) {
    if(ColaLlena()==1) return 0;
    Final=(Final+1)%MAXCOLA;
    EspacioCola[Final]=Dato;
    return 1;
}

template<class T>
int ColaCircular<T>::Get(void) {
    int Val;
    if(ColaVacía()==1) return 0;
    Frente=(Frente+1)%MAXCOLA;
    Val=EspacioCola[Frente];
    return Val;
}

template<class T>
void ColaCircular<T>::VerCola(void)const {
    int ini,fin;
    ini=Frente;
    fin=Final;
    if(ini==fin) cout<<endl<<"No hay elementos que mostrar.";
    else {
        if(ini!=fin) {
            cout<<endl<<"Estado de la cola: ";
            while(ini!=fin) {
                ini=(ini+1)%MAXCOLA;
                cout<<EspacioCola[ini]<<" ";
            }
        }
    }
}

int main(void) {
    ColaCircular<int> ObjCola;
    char opc;
    int dato,band;
    cout<<endl<<"Implementacion de una cola estatica";
    do {
```

```

cout<<endl<<endl<<"Menu de opciones:";
cout<<endl<<"1.- Insertar (Put)";
cout<<endl<<"2.- Eliminar (Get)";
cout<<endl<<"3.- Ver";
cout<<endl<<"4.- Salir"<<endl;
cout<<endl<<"Elija una opcion: ";
cin>>opc;
switch(opc) {
    case '1': cout<<endl<<"Introduce un valor numerico: ";
               cin>>dato;
               band=ObjCola.Put(dato);
               if(band==1) cout<<"El dato se ha insertado.";
               else if(band==0) cout<<"El dato no se pudo insertar, cola llena.";
               break;
    case '2': dato=ObjCola.Get();
               if(dato!=0) cout<<endl<<"Se quito el:"<<dato;
               else if(dato==0) cout<<"Cola vacia.";
               break;
    case '3': ObjCola.VerCola();
               break;
    case '4': cout<<"El programa ha terminado."<<endl;
               break;
}
} while(opc!='4');
}

```

5. Ahora localice el icono de **Compilar** que está al inicio de la barra de herramientas y oprímalo, proporcione el nombre de **ejercicio 15A** y elija el destino que guste para guardarlo.
6. Ahora ejecútelo y pruebe las operaciones de la cola.
7. En el menú **Archivo** seleccione la opción **Nuevo** y a continuación **Archivo Fuente**.
8. En el nuevo documento transcriba el código del siguiente programa, en el cual se construye una cola dinámica mediante la implementación de una lista, el código es prácticamente el mismo de la lista simple, hemos renombrado al apuntador **le**, que indicaba el inicio de la lista y ahora le llamamos **Ini**, además incluimos otro apuntador al final de la cola (**Fin**), que facilita la inserción al final del método **Put**, sin tener que recorrer toda la lista.

```

#include<iostream>
#include<iomanip>
using namespace std;

template<class T>
class Nodo {
public: T info;
       Nodo<T> *sig;
       Nodo( const T);

```

```

};

template<class T>
class Cola {
    private: Nodo<T> *Ini,*Fin;

    public: Cola();
           ~Cola();
           void Put(const T);
           T Get();
           void Ver();
           int Vacia();
};

template<class T>
Nodo<T>::Nodo(const T v) {
    info = v;
    sig = NULL;
}

template<class T>
Cola<T>::Cola() {
    Ini= Fin = NULL;
}

template<class T>
Cola<T>::~~Cola() {
    Nodo <T> *sale;
    int val;
    while(Ini) {
        sale=Ini;
        Ini=Ini->sig;
        val=sale->info;
        delete(sale);
        cout<<"Bloque de memoria liberado: "<<val<<endl;
    }
    delete(Ini);
}

template<class T>
void Cola<T>::Put(const T v) {
    Nodo <T>*nvo = new Nodo<T>(v);
    if(Vacia()) Ini=Fin=nvo;
    else {
        Fin->sig=nvo;
        Fin=nvo;
    }
}

```

```

    }
}

template<class T>
T Cola<T>::Get() {
    T val;
    Nodo<T> *sale = Ini;
    if(Vacia()) val= 0;
    else {
        val=sale->info;
        if(Ini==Fin) Ini=Fin=NULL;
        else Ini=sale->sig;
        delete sale;
    }
    return val;
}

template<class T>
int Cola<T>::Vacia() {
    return (Ini==NULL && Fin==NULL) ?1 :0;
}

template<class T>
void Cola<T>::Ver() {
    Nodo <T>*tmp = Ini;
    if(Vacia()) cout<<endl<<"Lista Cola esta vacia.";
    else {
        cout << "\nLa cola es: ";
        while (tmp) {
            cout <<setw(4)<< tmp->info;
            tmp = tmp->sig;
        }
    }
    cout <<endl;
}

int main(void) {
    Cola <int> cola;
    char opc;
    int valor;
    cout<<endl<<"\nImplementacion de una Cola mediante una Lista"<<endl;
    do {
        cout<<"\n1.- Insertar (Put)\n2.- Eliminar (Get)\n3.- Ver\n4.- Salir\n\n";
        cout<<"Opcion: ";
        cin>>opc;
        switch(opc) {

```

```
        case '1': cout<<endl<<"Introduce un valor diferente de cero: ";
                  cin>>valor;
                  cola.Put(valor);
                  break;
        case '2': valor=cola.Get();
                  if(valor==0) cout<<endl<<"La Cola esta vacia." <<endl;
                  else cout<<endl<<"El dato quitado es: "<<valor<<endl;
                  break;
        case '3': cola.Ver();
                  break;
    }
} while(opc!='4');
}
```

9. Ahora localice el icono de **Compilar** que está al inicio de la barra de herramientas y oprímalo, proporcione el nombre de **ejercicio 15B** y elija el destino que guste para guardarlo.
10. Ahora ejecútelo y pruebe las operaciones de su cola.
11. Fin de la Práctica.