Universidad De San Carlos de Guatemala
Centro Universitario de Occidente CUNOC


# MANUAL DE TÉCNICO

202031773 Enmer Oswaldo Sandoval Mazariegos

# Nodo

Juego.h | Nodo.h

```
//
// Created by laptop on 8/03/2024.
//

#ifndef PRACTICA1EDD_NODO_H
#define PRACTICA1EDD_NODO_H


#include "Carta.h"

class Nodo {
public:
    Carta* carta;
    Nodo* siguiente;

    Nodo(Carta* carta);
};


#endif //PRACTICA1EDD_NODO_H
```

Juego.h | Nodo.h | Nodo.cpp

```
//
// Created by laptop on 8/03/2024.
//

#include "Nodo.h"

Nodo::Nodo(Carta *carta) : carta(carta), siguiente(nullptr) {}
```

# Clase carta

Juego.h | Juego.cpp | Carta.h

```
11  public:
12      int numero;
13      Palo palo;
14      char color;
15      bool estado;
16
17      Carta(int numero, Palo palo, char color, bool estado);
18
19      int getNumer();
20      Palo getPalo();
21      char getColor();
22      bool getEstado();
23
24      void setNumero(int numero);
25      void setPalo(Palo palo);
26      void setColor(char color);
27      void setEstado(bool estado);
28  };
29
30
31  #endif //PRACTICA1EDD_CARTA_H
32
```

Juego.h | Juego.cpp | Carta.h | Carta.cpp

```
1   //
2   // Created by laptop on 8/03/2024.
3   //
4
5   #include "Carta.h"
6
7   Carta::Carta(int numero, Palo palo, char color, bool estado) {
8       this->numero = numero;
9       this->palo = palo;
10      this->color = color;
11      this->estado = estado;
12  }
13
14  void Carta::setNumero(int numero) {
15      this->numero = numero;
16  }
17
18  void Carta::setPalo(Palo palo) {
19      this->palo = palo;
20  }
21
22  void Carta::setColor(char color) {
23      this->color = color;
24  }
25
26  void Carta::setEstado(bool estado) {
27      this->estado = estado;
```

# Pila

Juego.h | Juego.cpp | Carta.h | Carta.cpp | Pila.cpp

```cpp
#include ...

Pila::Nodo::Nodo(Carta* carta) : carta(carta), siguiente(nullptr) {}

Pila::Pila() : cima(nullptr) {}

void Pila::push(Carta* carta) {
    Nodo* nodo = new Nodo(carta);
    nodo->siguiente = cima;
    cima = nodo;
}

void Pila::pop() {
    if (cima != nullptr) {
        Nodo* nodoAEliminar = cima;
        cima = cima->siguiente;
        delete nodoAEliminar;
    }
}

Carta* Pila::top() {
    if (cima != nullptr) {
        return cima->carta;
    } else {
        return nullptr;
    }
```

Juego.h | Juego.cpp | Carta.h | Carta.cpp | Pila.h

```cpp
#define PRACTICAEDD_PILA_H

#include ...

class Pila {
public:
    class Nodo {
        Carta* carta;
        Nodo* siguiente;

        Nodo(Carta* carta);
    };

    Nodo* cima;

    Pila();

    void push(Carta* carta);
    void pop();
    Carta* top();
    bool estaVacia();
};
```

# Colas

Juego.h | Juego.cpp | Carta.h | Carta.cpp | Pila.cpp | ColasPrincipales.h | ColasPrincipales.cpp

```
class ColasPrincipales {
public:
    class Nodo {

    public:
        Nodo(Carta* carta);

        Nodo* siguiente;
        Carta* carta;
    };

    Nodo* frente;
    Nodo* final;

    ColasPrincipales();

    void encolar(Carta* carta);
    void desencolar();
    Carta* peek();
    bool estaVacia();
};

#endif //PRACTICA1EDD_COLASPRINCIPALES_H
```

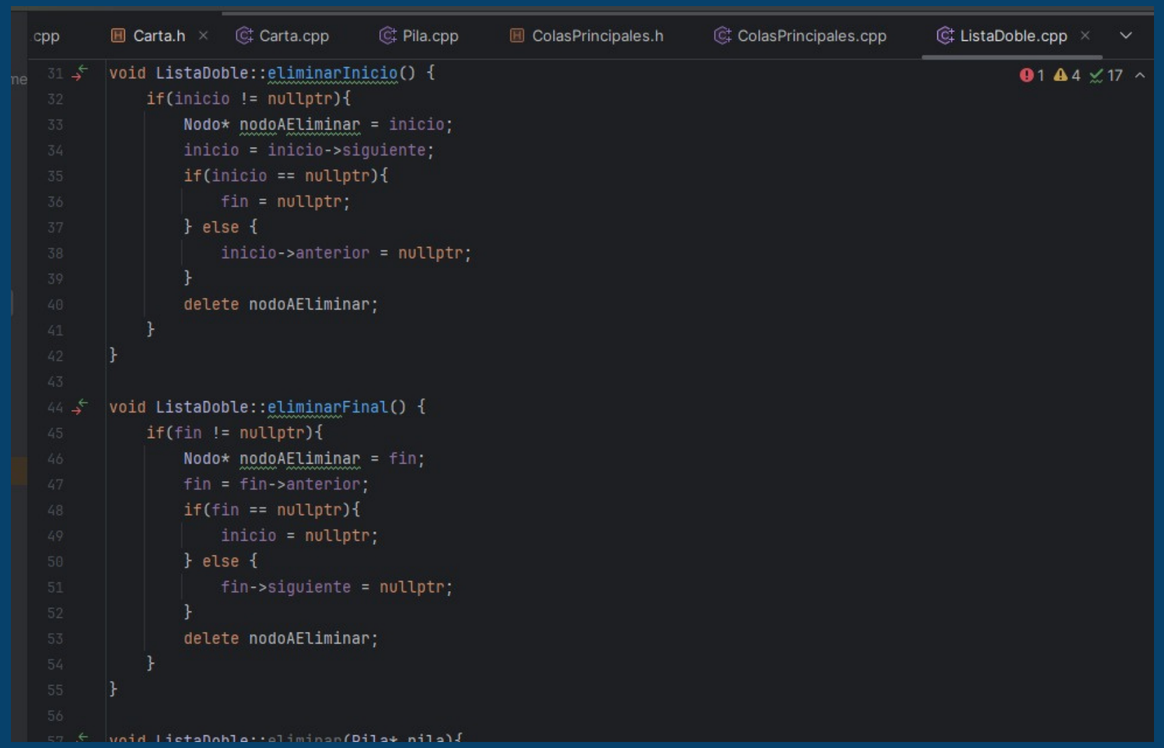
Juego.h | Juego.cpp | Carta.h | Carta.cpp | Pila.cpp | ColasPrincipales.h | ColasPrincipales.cpp

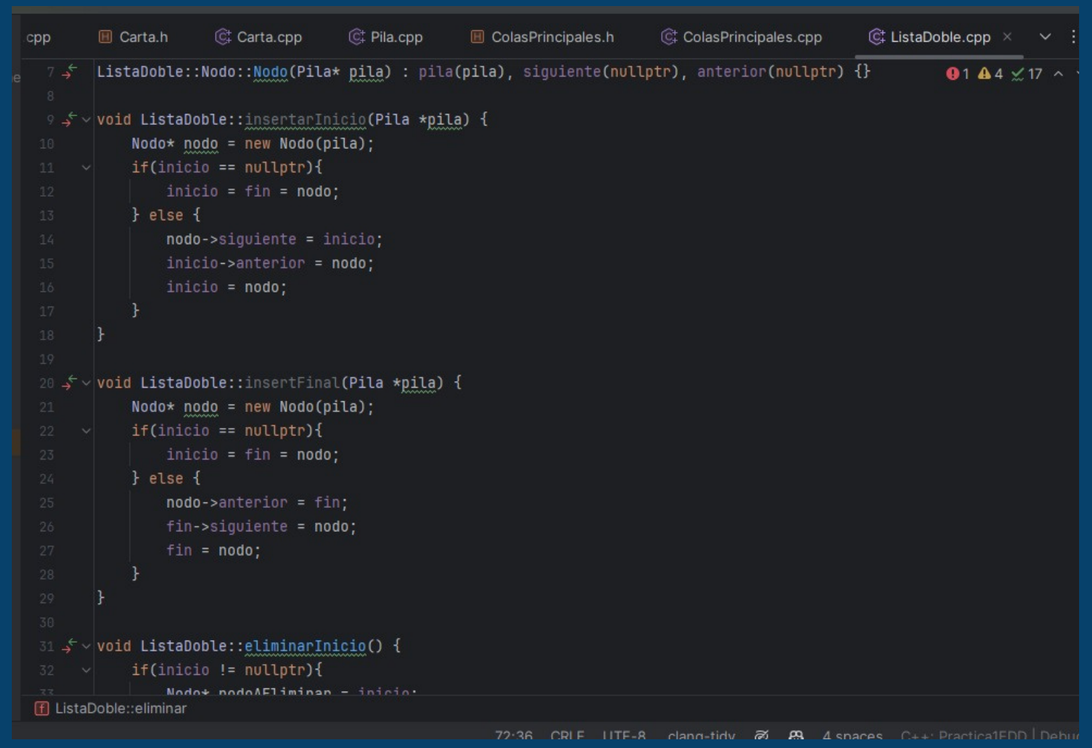
```
void ColasPrincipales::encolar(Carta *carta) {
    Nodo* nodo = new Nodo(carta);
    if(estaVacia()){
        frente = final = nodo;
    } else {
        final->siguiente = nodo;
        final = nodo;
    }
}

void ColasPrincipales::desencolar() {
    if(!estaVacia()){
        Nodo* nodoAEliminar = frente;
        frente = frente->siguiente;
        if(frente == nullptr){
            final = nullptr;
        }
        delete nodoAEliminar;
    }
}

Carta* ColasPrincipales::peek() {
    if (!estaVacia()){
        return frente->carta;
    } else{
```

# Lista Doblemente enlazada



# Lista Doblemente enlazada