



Universidad Autónoma de Santo Domingo (UASD)

Nombre: Ismael Bernardo Nuñez Feliz

Matrícula: 100284129

Asignatura: INF5260 Estructura De Datos

Sección: Z01

Maestro Asignado: Silverio Del Orbe Abad

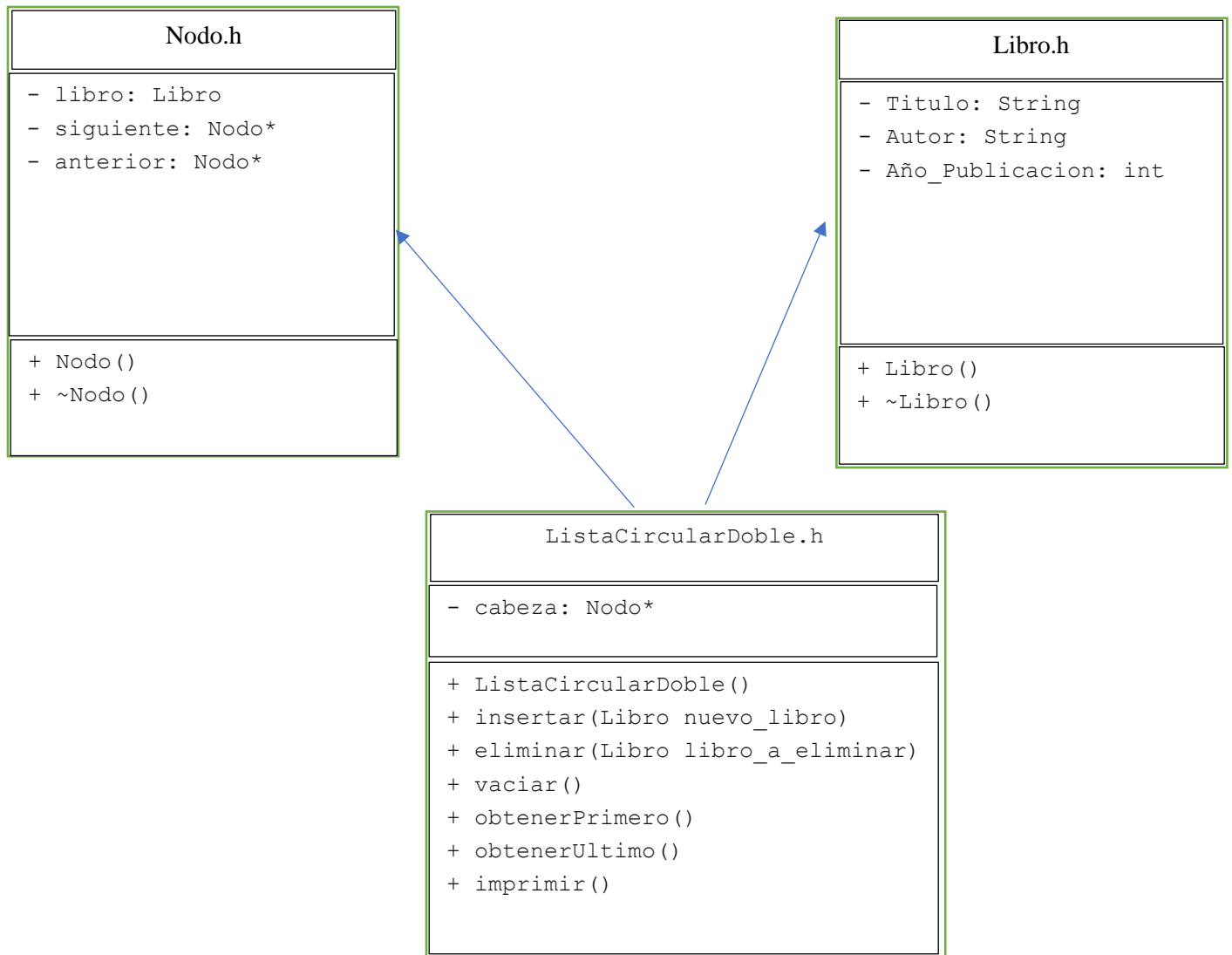
Tema: Domingo, 15 de octubre de 2023

Fecha de Entrega: Caso práctico 2.3: Implemente en C++ una TAD Lineal

Contenido

- Diseño UML 1
- Código 2
 - Libro.h 2
 - Nodo.h 2
 - ListaCirculaDoble.h..... 2
 - ListaCirculaDoble.cpp 3
 - Main.cpp 5
- ScreenShots 8
- 8

Diseño UML



Código

Libro.h

```
#ifndef LIBRO_H
#define LIBRO_H

#include <string>

class Libro {
public:
    std::string titulo;
    std::string autor;
    int anio_publicacion;
};

#endif
```

Nodo.h

```
#ifndef NODO_H
#define NODO_H

#include "Libro.h"

class Nodo {
public:
    Libro libro;
    Nodo* siguiente;
    Nodo* anterior;
};

#endif
```

ListaCirculaDoble.h

```
#ifndef LISTACIRCULARDOBLE_H
#define LISTACIRCULARDOBLE_H

#include "Nodo.h"

class ListaCircularDoble {
public:
    Nodo* cabeza;
    ListaCircularDoble();
    void insertar(Libro nuevo_libro);
};
```

```

    void eliminar(Libro libro_a_eliminar);
    void vaciar();
    Libro obtenerPrimero();
    Libro obtenerUltimo();
    void imprimir();
};

#endif

```

ListaCirculaDoble.cpp

```

#include "ListaCircularDoble.h"

#include <iostream>

ListaCircularDoble::ListaCircularDoble() {
    cabeza = NULL;

    Libro libro1 = {"El gran Gatsby", "F. Scott Fitzgerald", 1925};
    Libro libro2 = {"Cien años de soledad", "Gabriel García Márquez", 1967};
    Libro libro3 = {"1984", "George Orwell", 1949};

    insertar(libro1);
    insertar(libro2);
    insertar(libro3);
}

void ListaCircularDoble::insertar(Libro nuevo_libro) {
    Nodo* nuevo_nodo = new Nodo();
    nuevo_nodo->libro = nuevo_libro;
    nuevo_nodo->siguiente = cabeza;
    nuevo_nodo->anterior = NULL;

    if (cabeza != NULL) {
        cabeza->anterior = nuevo_nodo;
        Nodo* ultimo_nodo = cabeza->anterior;
        ultimo_nodo->siguiente = nuevo_nodo;
    } else {
        nuevo_nodo->anterior = nuevo_nodo;
    }

    cabeza = nuevo_nodo;
}

void ListaCircularDoble::eliminar(Libro libro_a_eliminar) {
    Nodo* nodo_actual = cabeza;

```

```

        do {
            if (nodo_actual->libro.titulo == libro_a_eliminar.titulo &&
nodo_actual->libro.autor == libro_a_eliminar.autor && nodo_actual-
>libro.anio_publicacion == libro_a_eliminar.anio_publicacion) {
                if (nodo_actual == cabeza) {
                    cabeza = nodo_actual->siguiente;
                }

                if (nodo_actual->anterior != NULL) {
                    nodo_actual->anterior->siguiente = nodo_actual->siguiente;
                }

                if (nodo_actual->siguiente != NULL) {
                    nodo_actual->siguiente->anterior = nodo_actual->anterior;
                }

                delete nodo_actual;

                return;
            }

            nodo_actual = nodo_actual->siguiente;
        } while (nodo_actual != cabeza);
    }

void ListaCircularDoble::vaciar() {
    Nodo* nodo_actual = cabeza;

    while (nodo_actual != NULL) {
        Nodo* siguiente_nodo = nodo_actual->siguiente;

        delete nodo_actual;

        nodo_actual = siguiente_nodo;

        if (nodo_actual == cabeza) {
            break;
        }
    }

    cabeza = NULL;
}

Libro ListaCircularDoble::obtenerPrimero() {
    return cabeza->libro;
}

```

```

Libro ListaCircularDoble::obtenerUltimo() {
    return cabeza->anterior->libro;
}

void ListaCircularDoble::imprimir() {
    Nodo* nodo_actual = cabeza;

    if (cabeza != NULL) {
        std::cout << "Lista de libros:" << std::endl << std::endl;

        do {
            std::cout << "Titulo: " << nodo_actual->libro.titulo <<
std::endl;
            std::cout << "Autor: " << nodo_actual->libro.autor << std::endl;
            std::cout << "Año de publicacion: " << nodo_actual->libro.ano_publicacion << std::endl << std::endl;

            nodo_actual = nodo_actual->siguiente;
        } while (nodo_actual != cabeza);

        std::cout << std::endl;
    } else {
        std::cout << "La lista está vacía." << std::endl << std::endl;
    }
}

```

Main.cpp

```

#include <iostream>
#include "ListaCircularDoble.h"

int main() {
    ListaCircularDoble lista_libros;

    int opcion = 0;

    lista_libros.imprimir();
    while (opcion < 6) {

        std::cout << "Seleccione una opcion:" << std::endl;
        std::cout << "1. Insertar" << std::endl;
        std::cout << "2. Eliminar" << std::endl;
        std::cout << "3. Vaciar" << std::endl;
        std::cout << "4. Obtener primero" << std::endl;
        std::cout << "5. Obtener ultimo" << std::endl;
    }
}

```

```

std::cout << "6. Salir" << std::endl;

std::cin >> opcion;

switch (opcion) {
    case 1: {
        Libro nuevo_libro;
        std::cout << "Ingrese el titulo del libro: ";
        std::cin >> nuevo_libro.titulo;
        std::cout << "Ingrese el autor del libro: ";
        std::cin >> nuevo_libro.autor;
        std::cout << "Ingrese el año de publicacion del libro: ";
        std::cin >> nuevo_libro.anio_publicacion;

        lista_libros.insertar(nuevo_libro);

        lista_libros.imprimir();

        break;
    }
    case 2: {
        Libro libro_a_eliminar;
        std::cout << "Ingrese el titulo del libro a eliminar: ";
        std::cin >> libro_a_eliminar.titulo;
        std::cout << "Ingrese el autor del libro a eliminar: ";
        std::cin >> libro_a_eliminar.autor;
        std::cout << "Ingrese el año de publicacion del libro a
eliminar: ";
        std::cin >> libro_a_eliminar.anio_publicacion;

        lista_libros.eliminar(libro_a_eliminar);

        lista_libros.imprimir();

        break;
    }
    case 3: {
        lista_libros.vaciar();

        lista_libros.imprimir();

        break;
    }
    case 4: {
        Libro primer_libro = lista_libros.obtenerPrimero();

```



```
        std::cout << "El primer libro en la lista es:" << std::endl;
        std::cout << "Titulo: " << primer_libro.titulo << ", Autor: "
<< primer_libro.autor << ", Año de publicacion: " <<
primer_libro.anio_publicacion << "." << std::endl;

        break;
    }
    case 5: {
        Libro ultimo_libro = lista_libros.obtenerUltimo();

        std::cout << "El ultimo libro en la lista es:" << std::endl;
        std::cout << "Titulo: " << ultimo_libro.titulo << ", Autor: "
<< ultimo_libro.autor << ", Año de publicacion: " <<
ultimo_libro.anio_publicacion << "." << std::endl;

        break;
    }
    case 6:
        break;
    default:
        std::cout << "Opcion invalida." << std::endl;

        break;
    }

}

return 0;
}
```

ScreenShots

```

PS C:\Users\Ismael\Documents\Uasd2023-2> cd estructuraDedatos/unidad2
PS C:\Users\Ismael\Documents\Uasd2023-2\estructuraDedatos\unidad2> g++ ListaCircularDoble.cpp main.cpp -o main
PS C:\Users\Ismael\Documents\Uasd2023-2\estructuraDedatos\unidad2> ./main.exe
Seleccione una opcion:
1. Insertar
2. Eliminar
3. Vaciar
4. Obtener primero
5. Obtener ultimo
6. Salir
1
Ingrese el titulo del libro: codigoDavinci
Ingrese el autor del libro: danBrown
Ingrese el año de publicacion del libro: 1987
Lista de libros:

Titulo: codigoDavinci
Autor: danBrown
Año de publicacion: 1987

Seleccione una opcion:
1. Insertar
2. Eliminar
3. Vaciar
4. Obtener primero
5. Obtener ultimo
6. Salir
4
El primer libro en la lista es:
Titulo: codigoDavinci, Autor: danBrown, Año de publicacion: 1987.
Seleccione una opcion:
1. Insertar
2. Eliminar
3. Vaciar
4. Obtener primero
5. Obtener ultimo
6. Salir

```

