UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA

SEDE SOLOLÁ, SOLOLÁ, INGENIERÍA EN SISTEMAS

TERCER CICLO

Catedra: Programación I.

Catedrático: Inge. Neftalí Cristian Garcia López

Tema:

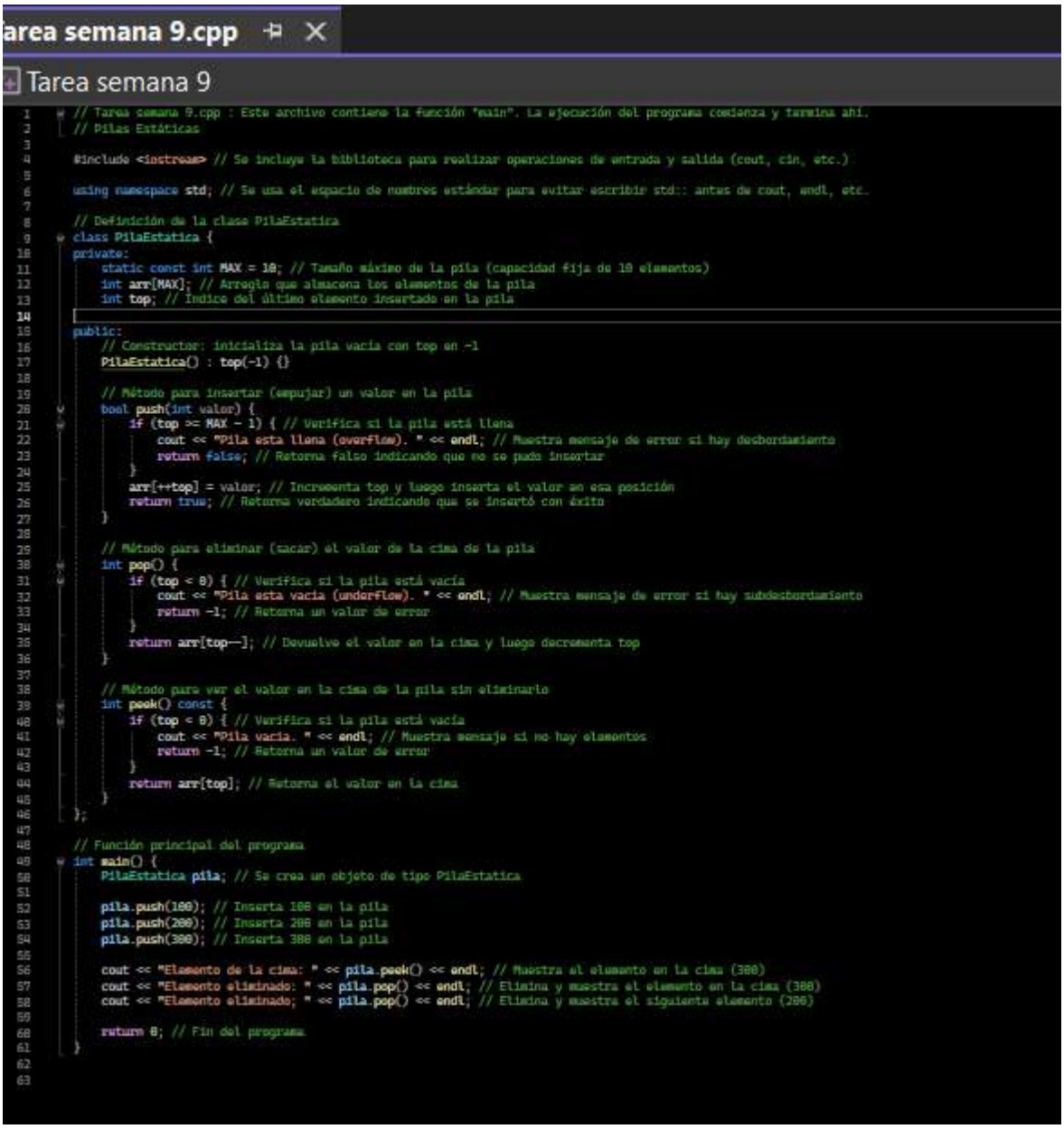
Tarea semana 9

Estudiante:

Selvin Geovany Tzunún Balán. Carnet: 2290-24-44.

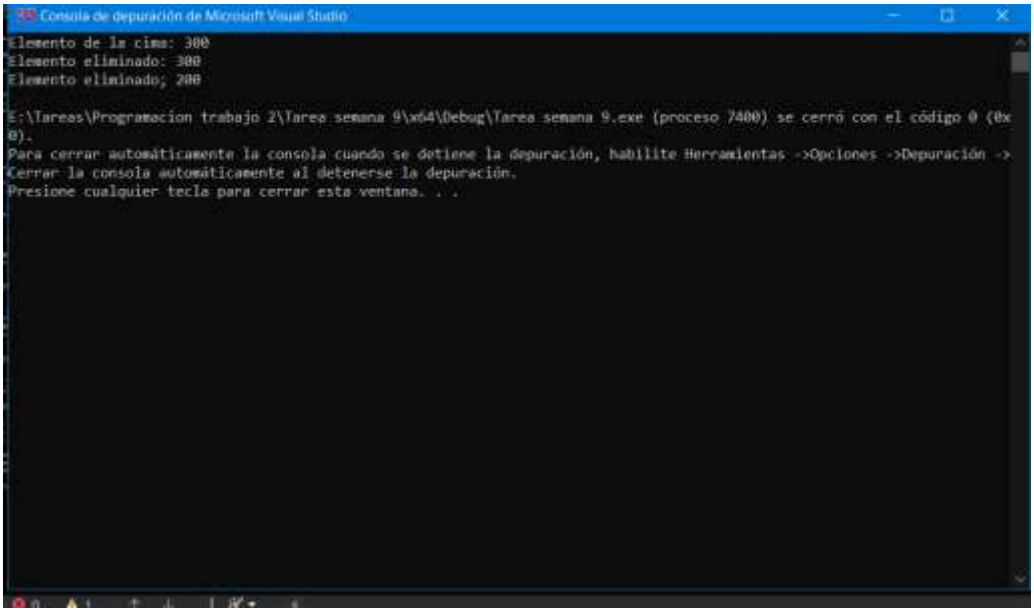
Pascual Evelio Yaxon Tziquin Carnet: 2290-24-26037

Pilas Estáticas: Código Comentado:

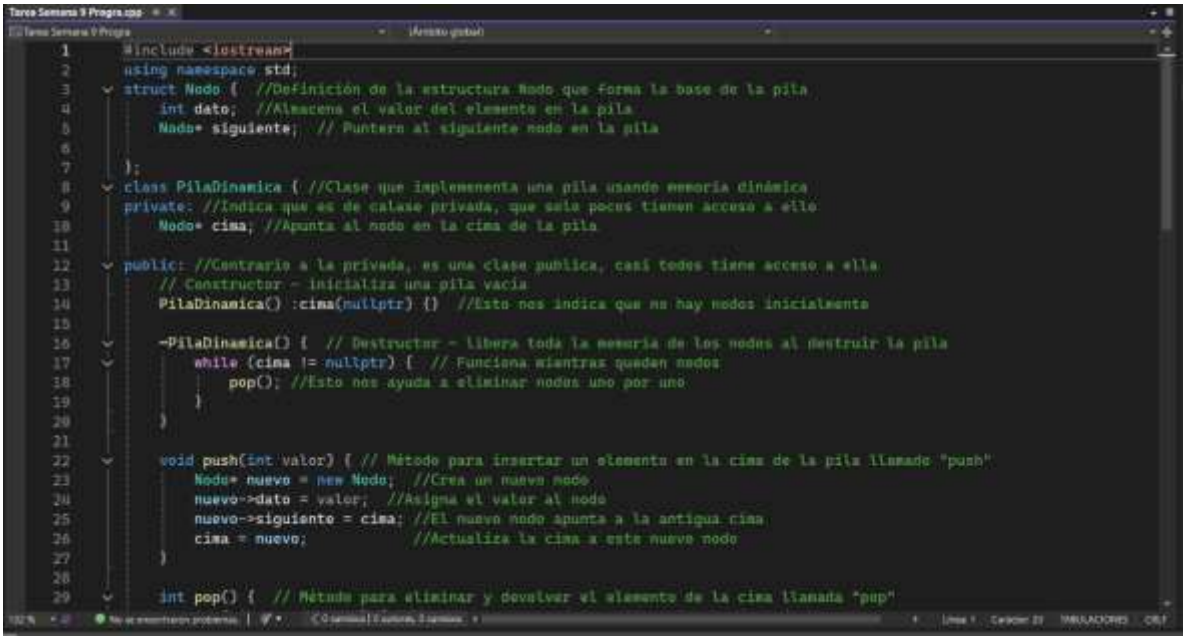


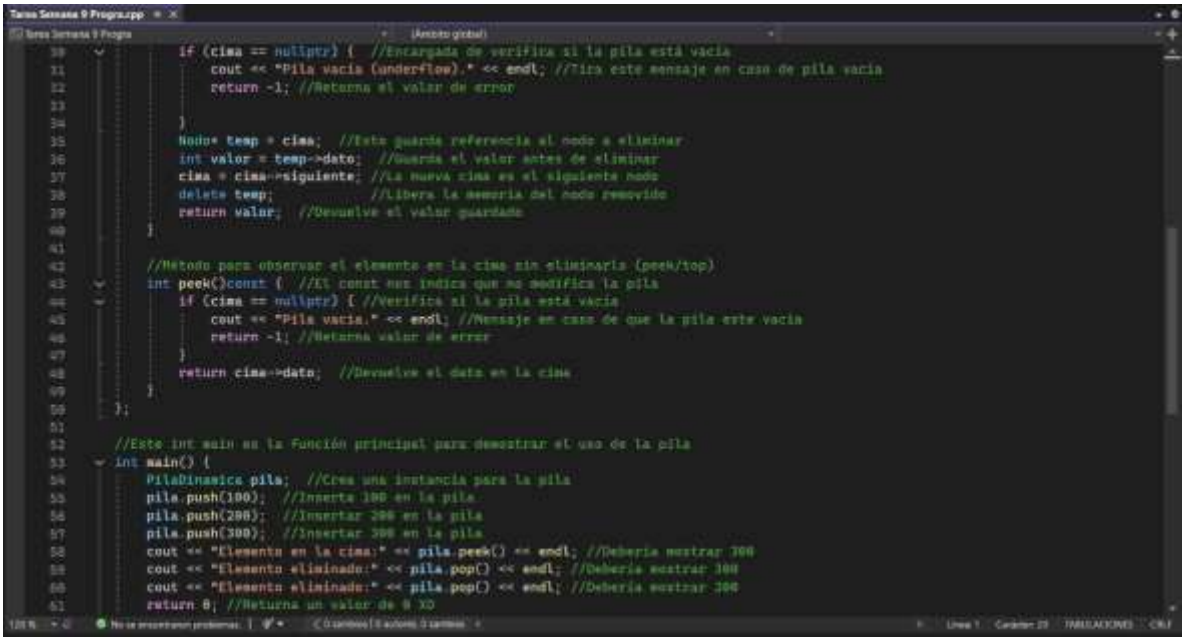

```cpp
// Tarea semana 9.cpp : Este archivo contiene la función "main". La ejecución del programa comienza y termina ahí.
// Pilas Estáticas

#include <iostream> // Se incluye la biblioteca para realizar operaciones de entrada y salida (cout, cin, etc.)

using namespace std; // Se usa el espacio de nombres estándar para evitar escribir std:: antes de cout, endl, etc.

// Definición de la clase PilaEstatica
class PilaEstatica {
private:
    static const int MAX = 10; // Tamaño máximo de la pila (capacidad fija de 10 elementos)
    int arr[MAX]; // Arreglo que almacena los elementos de la pila
    int top; // Índice del último elemento insertado en la pila

public:
    // Constructor: inicializa la pila vacía con top en -1
    PilaEstatica() : top(-1) {}

    // Método para insertar (empujar) un valor en la pila
    bool push(int valor) {
        if (top >= MAX - 1) { // Verifica si la pila está llena
            cout << "Pila esta llena (overflow). " << endl; // Muestra mensaje de error si hay desbordamiento
            return false; // Retorna falso indicando que no se pudo insertar
        }
        arr[++top] = valor; // Incrementa top y luego inserta el valor en esa posición
        return true; // Retorna verdadero indicando que se insertó con éxito
    }

    // Método para eliminar (sacar) el valor de la cima de la pila
    int pop() {
        if (top < 0) { // Verifica si la pila está vacía
            cout << "Pila esta vacia (underflow). " << endl; // Muestra mensaje de error si hay subdesbordamiento
            return -1; // Retorna un valor de error
        }
        return arr[top--]; // Devuelve el valor en la cima y luego decrementa top
    }

    // Método para ver el valor en la cima de la pila sin eliminarlo
    int peek() const {
        if (top < 0) { // Verifica si la pila está vacía
            cout << "Pila vacia. " << endl; // Muestra mensaje si no hay elementos
            return -1; // Retorna un valor de error
        }
        return arr[top]; // Retorna el valor en la cima
    }
};

// Función principal del programa
int main() {
    PilaEstatica pila; // Se crea un objeto de tipo PilaEstatica

    pila.push(100); // Inserta 100 en la pila
    pila.push(200); // Inserta 200 en la pila
    pila.push(300); // Inserta 300 en la pila

    cout << "Elemento de la cima: " << pila.peek() << endl; // Muestra el elemento en la cima (300)
    cout << "Elemento eliminado: " << pila.pop() << endl; // Elimina y muestra el elemento en la cima (300)
    cout << "Elemento eliminado: " << pila.pop() << endl; // Elimina y muestra el siguiente elemento (200)

    return 0; // Fin del programa
}
```

Compilación



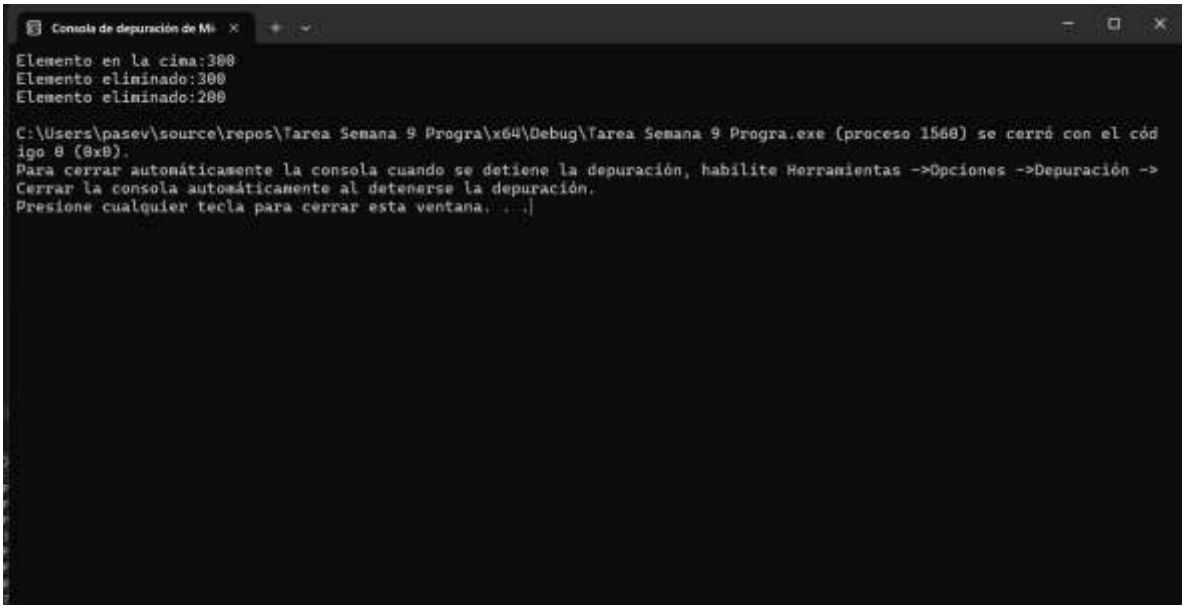
Pila Dinámica, Código Comentado:





```
        if (cima == nullptr) {  //Encargada de verifica si la pila está vacia
            cout << "Pila vacia (underflow)." << endl; //Tira este mensaje en caso de pila vacia
            return -1; //Retorna el valor de error

        }
        Nodo* temp = cima;  //Este guarda referencia al nodo a eliminar
        int valor = temp->dato;  //Guarda el valor antes de eliminar
        cima = cima->siguiente; //La nueva cima es el siguiente nodo
        delete temp;            //Libera la memoria del nodo removido
        return valor;  //Devuelve el valor guardado
    }

    //Método para observar el elemento en la cima sin eliminarla (peek/top)
    int peek()const {  //El const nos indica que no modifica la pila
        if (cima == nullptr) { //Verifica si la pila está vacia
            cout << "Pila vacia." << endl; //Mensaje en caso de que la pila este vacia
            return -1; //Retorna valor de error
        }
        return cima->dato;  //Devuelve el dato en la cima
    }
};

//Este int main es la función principal para demostrar el uso de la pila
int main() {
    PilaDinamica pila;  //Crea una instancia para la pila
    pila.push(100);  //Inserta 100 en la pila
    pila.push(200);  //Insertar 200 en la pila
    pila.push(300);  //Insertar 300 en la pila
    cout << "Elemento en la cima:" << pila.peek() << endl; //Deberia mostrar 300
    cout << "Elemento eliminado:" << pila.pop() << endl; //Deberia mostrar 300
    cout << "Elemento eliminado:" << pila.pop() << endl; //Deberia mostrar 200
    return 0; //Returna un valor de 0 XD
```

Compilación:



```
Consola de depuración de Mi
Elemento en la cima:300
Elemento eliminado:300
Elemento eliminado:200

C:\Users\pasev\source\repos\Tarea Semana 9 Progra\x64\Debug\Tarea Semana 9 Progra.exe (proceso 1560) se cerró con el cód
igo 0 (0x0).
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```

Link del Repositorio:

https://github.com/E-lev10/Tarea-Semana-9-progra.git