18-3-2021

Lista estática

Practica: 2

Materia: Seminario de estructura de datos 1

Sección: D13.

Código: 216584703

Carrera: Ingeniería en computación.

Nombre alumno: Padilla Pérez Jorge Daray

Nombre profesor: Julio Esteban Valdes Lopez



Índice general

Índice **1**

**Introducción 2**

**Pantallazos 3-6**

**Conclusión 7**

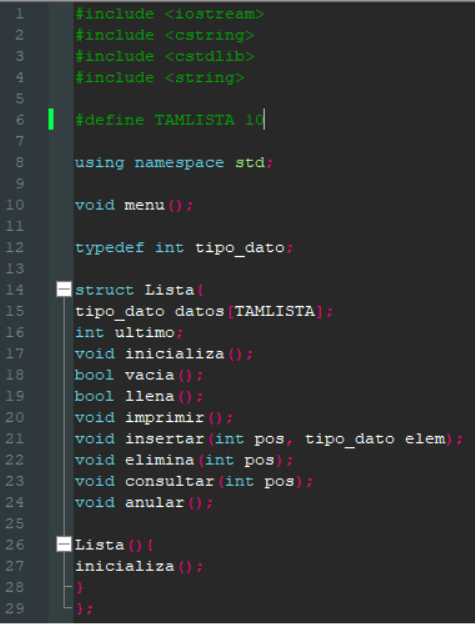
**Código fuente 8-13**

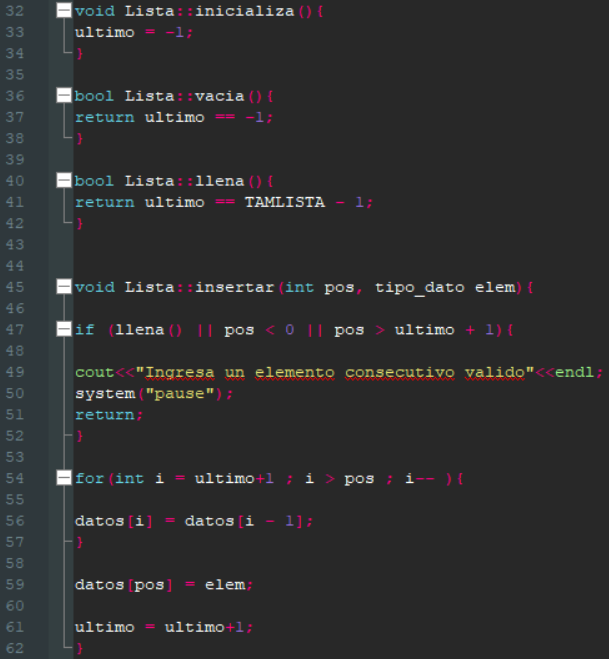
Introducción

Mi práctica consiste en la implementación de una lista estática en lenguaje c++, en donde se puede apreciar la integración de la frase: divide y vencerás, además de introducir el tema de las listas y el usar posiciones para agregar y quitar cosas, mediante estos procesos se tomó en cuenta la implementación de una función de insertar, eliminar y mostrar.

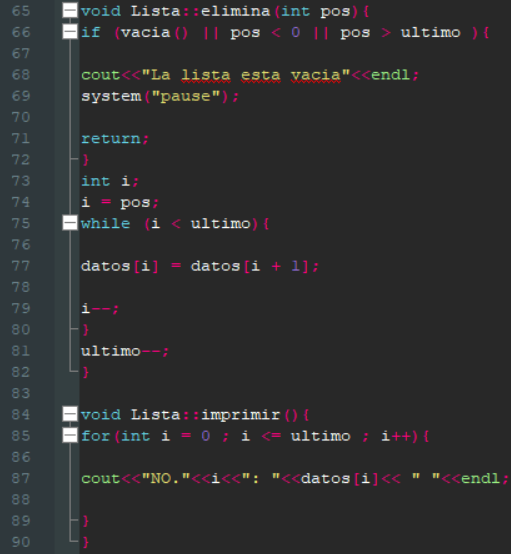
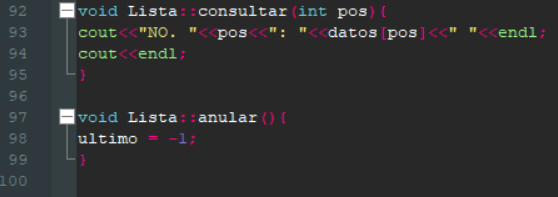
Al realizar mi practica obtuve como resultado la correcta integración e implementación de la lista estática dentro del código, en el cual es necesario asegurar que cada llamada a la lista debe estar definida sobre un problema menos complejo que el que dio lugar a la llamada, , como el que yo diseñe, al cual hago referencia en la página 12 logrando obtener un mejor conocimiento del tema.

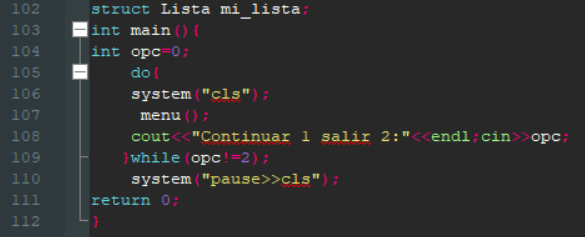
Pantallazo

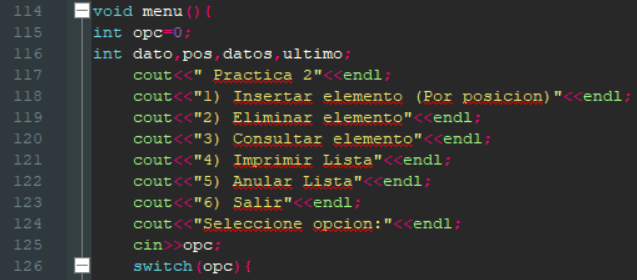
aquí se aprecia la inicialización de las funciones de forma diferente a como venimos manejando pero que funciona de buena manera también.



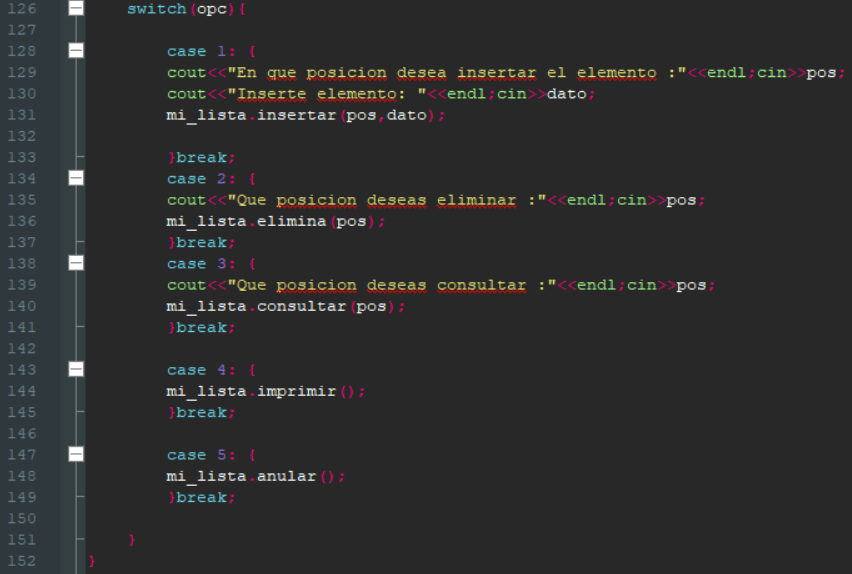
Aquí se mira donde se les dan los parámetros adecuados a las funciones para su correcto funcionamiento.

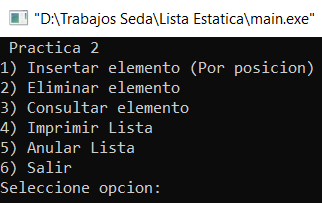


Implementación de la variable mi\_lista derivada de la estructura Lista para mejor entendimiento en código después , además de implementar el menú y un sistema para preguntar si se desea salir del programa o continuar para no cerrar el programa de golpe o por error.

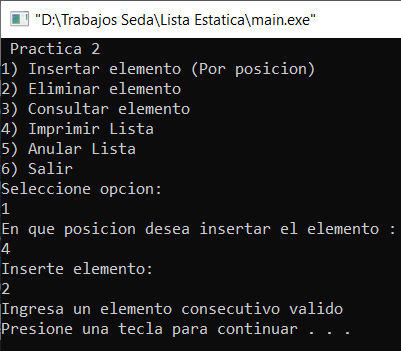


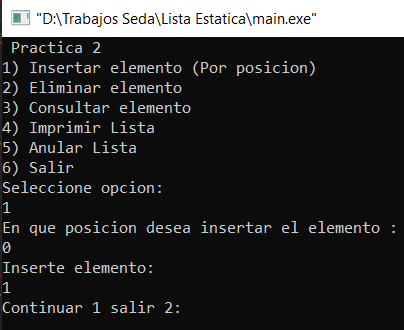
Aquí es donde ya se implementa el menú que vera el usuario mas el inicio de un switch para que se mande a llamar mejor lo que ordene el usuario

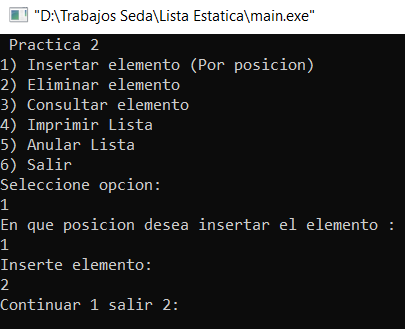
Aquí como lo dejamos se inicializa el switch con sus respectivas opciones en las cuales dependiendo lo que ingrese el usuario abrirá una opción u otra, en la cual simplemente se mandan a llamar las funciones anteriormente hechas con el simple hecho de mandar a pedir información y guardarla de manera adecuada.

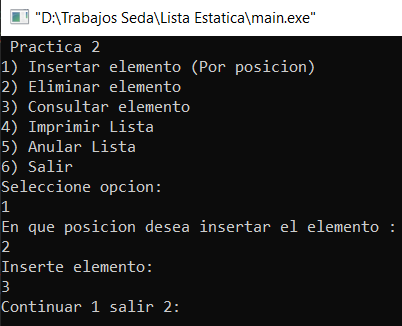


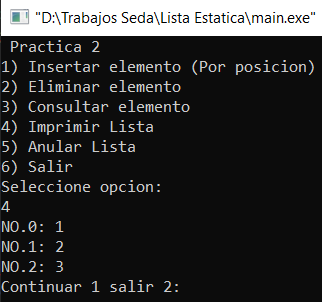
Aquí se aprecia el menú que debe tener el cual ve el usuario final



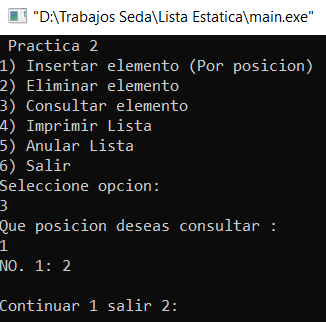
Aquí insertamos el primer elemento posición 0 y es el elemento llamado 1 ya que si ponemos una posición que no siga con la secuencia da error como el que se muestra a la derecha



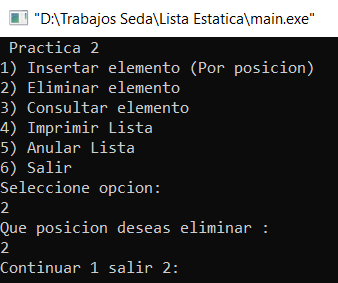
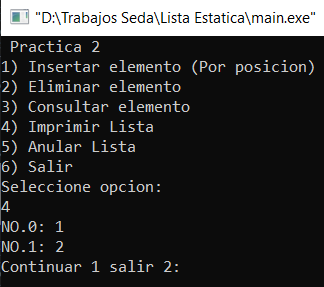
Entonces insertamos varios elementos como los de la izquierda para probar las otras funciones



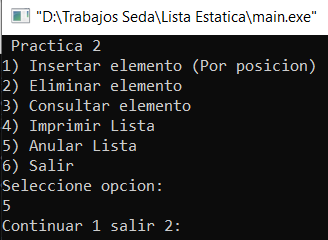
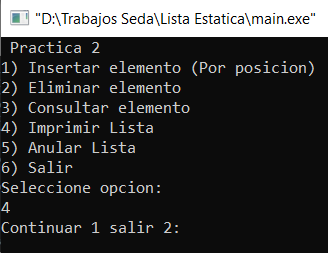
En primer lugar imprimimos los resultados con la opcion 4 en la cual se puede ver la posicion y el valor que tiene asignado.



Ahora consultamos un elemento, el cual se hace mediante posición y dándole la posición te muestra el elemento como se aprecia ahí



En este caso usamos la opción 2 eliminar y eliminamos la posición 2 en la cual estaba el 3 entonces al imprimir los elementos ya no aparece como se aprecia a la derecha.



Por ultimo se selecciona la opción 5 que es anular y anula la lista lo que se puede apreciar mejor del lado derecho a la hora de imprimirlo no aparece nada porque ya fue anulada

Conclusión

Respecto a la realización del código concluyo que logré el entendimiento correcto del tema mediante la correcta implementación de esta, se entendió el tema de la lista estática tanto de su sintaxis, como la lógica para su implementación, también aprendí los fallos que puede dar a la hora de correr el código y por ende sus posibles soluciones.

Además de aprender la teoría necesaria para darme cuenta de lo funcional que es, y poder identificar cuando conviene utilizar la lista estática y si se puede resolver el problema utilizándola.

**Código Fuente**

**#include <iostream>**

**#include <cstring>**

**#include <cstdlib>**

**#include <string>**

**#define TAMLISTA 10**

**using namespace std;**

**void menu();**

**typedef int tipo\_dato;**

**struct Lista{**

**tipo\_dato datos[TAMLISTA];**

**int ultimo;**

**void inicializa();**

**bool vacia();**

**bool llena();**

**void imprimir();**

**void insertar(int pos, tipo\_dato elem);**

**void elimina(int pos);**

**void consultar(int pos);**

**void anular();**

**Lista(){**

**inicializa();**

**}**

**};**

**void Lista::inicializa(){**

**ultimo = -1;**

**}**

**bool Lista::vacia(){**

**return ultimo == -1;**

**}**

**bool Lista::llena(){**

**return ultimo == TAMLISTA - 1;**

**}**

**void Lista::insertar(int pos, tipo\_dato elem){**

**if (llena() || pos < 0 || pos > ultimo + 1){**

**cout<<"Ingresa un elemento consecutivo valido"<<endl;**

**system("pause");**

**return;**

**}**

**for(int i = ultimo+1 ; i > pos ; i-- ){**

**datos[i] = datos[i - 1];**

**}**

**datos[pos] = elem;**

**ultimo = ultimo+1;**

**}**

**void Lista::elimina(int pos){**

**if (vacia() || pos < 0 || pos > ultimo ){**

**cout<<"La lista esta vacia"<<endl;**

**system("pause");**

**return;**

**}**

**int i;**

**i = pos;**

**while (i < ultimo){**

**datos[i] = datos[i + 1];**

**i++;**

**}**

**ultimo--;**

**}**

**void Lista::imprimir(){**

**for(int i = 0 ; i <= ultimo ; i++){**

**cout<<"NO."<<i<<": "<<datos[i]<< " "<<endl;**

**}**

**}**

**void Lista::consultar(int pos){**

**cout<<"NO. "<<pos<<": "<<datos[pos]<<" "<<endl;**

**cout<<endl;**

**}**

**void Lista::anular(){**

**ultimo = -1;**

**}**

**struct Lista mi\_lista;**

**int main(){**

**int opc=0;**

**do{**

**system("cls");**

**menu();**

**cout<<"Continuar 1 salir 2:"<<endl;cin>>opc;**

**}while(opc!=2);**

**system("pause>>cls");**

**return 0;**

**}**

**void menu(){**

**int opc=0;**

**int dato,pos,datos,ultimo;**

**cout<<" Practica 2"<<endl;**

**cout<<"1) Insertar elemento (Por posicion)"<<endl;**

**cout<<"2) Eliminar elemento"<<endl;**

**cout<<"3) Consultar elemento"<<endl;**

**cout<<"4) Imprimir Lista"<<endl;**

**cout<<"5) Anular Lista"<<endl;**

**cout<<"6) Salir"<<endl;**

**cout<<"Seleccione opcion:"<<endl;**

**cin>>opc;**

**switch(opc){**

**case 1: {**

**cout<<"En que posicion desea insertar el elemento :"<<endl;cin>>pos;**

**cout<<"Inserte elemento: "<<endl;cin>>dato;**

**mi\_lista.insertar(pos,dato);**

**}break;**

**case 2: {**

**cout<<"Que posicion deseas eliminar :"<<endl;cin>>pos;**

**mi\_lista.elimina(pos);**

**}break;**

**case 3: {**

**cout<<"Que posicion deseas consultar :"<<endl;cin>>pos;**

**mi\_lista.consultar(pos);**

**}break;**

**case 4: {**

**mi\_lista.imprimir();**

**}break;**

**case 5: {**

**mi\_lista.anular();**

**}break;**

**}**

**}**