***ACTIVIDAD N°3 Ejercicio 1***

***Desarrollo De Un Programa De Gestión De Tareas***

***Informa***

***#include <iostream>***

***#include <vector>***

***#include <string>***

***using namespace std; :*** En las primeras 3 líneas se incluyen 3 bibliotecas para poder utilizar cosas como el ***VECTOR*** o la ***CADENA DE TEXTO.*** Mientras que la última se utiliza para no tener que escribir código de más.

***vector<string> listaTareas; :*** En esta línea se declara una lista de tipo ***STRING*** para almacenar las tareas que el usuario ingrese.

***void menuPrincipal();***

***void agregarTarea();***

***void eliminarTarea();***

***void mostrarTareas(); :*** En estas líneas se declaran las funciones que se usaran en el programa.

***int main() {***

***menuPrincipal();***

***return 0; } :*** Aquí, la función ***MAIN*** es el punto de entrada del programa y este llama a la función ***MENUPRINCIPAL***, el cual se encarga de mostrar el menú y gestionar la interacción con el usuario.

***void menuPrincipal() {***

***int opcion;***

***do {***

***cout << "\n--- Lista de Tareas ---" << endl;***

***cout << "1. Agregar tarea" << endl;***

***cout << "2. Eliminar tarea" << endl;***

***cout << "3. Mostrar tareas" << endl;***

***cout << "4. Salir" << endl;***

***cout << "Seleccione una opción: ";***

***cin >> opcion;***

***cin.ignore();***

***switch(opcion) {***

***case 1:***

***agregarTarea();***

***break;***

***case 2:***

***eliminarTarea();***

***break;***

***case 3:***

***mostrarTareas();***

***break;***

***case 4:***

***cout << "Saliendo..." << endl;***

***break;***

***default:***

***cout << "Opción inválida, por favor elija nuevamente." << endl;***

***}***

***} while(opcion != 4);***

***} :*** Todo este código hace varias cosas:

***.INT OPCION; :*** Es una variable que se usa para almacenar la opción que el usuario elige en el menú.

***.DO-WHILE LOOP:*** Esto muestra el menú de opciones y espera una selección del usuario, repitiéndose hasta que el usuario elija la opción ***4*** (salir).

***.SWITCH(OPCION) :*** Según la opción seleccionada, se llama a una de las funciones: ***AgregarTarea , EliminarTarea , o MostrarTareas***.

***void agregarTarea() {***

***string tarea;***

***cout << "Ingrese la tarea: ";***

***getline(cin, tarea);***

***listaTareas.push\_back(tarea);***

***cout << "Tarea '" << tarea << "' agregada." << endl;***

***} :*** Aquí suceden varias cosas:

***. STRING TAREA; :*** Declara una variable de tipo ***STRING*** para almacenar la nueva tarea.

***GETLINE(CIN, TAREA); :*** Lee una línea completa del usuario y la almacena en ***TAREA.***

***LISTATAREAS.PUSH\_BACK[(tarea); :*** Agrega la tarea ingresada al final de la lista ***LISTATAREAS.***

***void eliminarTarea() {***

***mostrarTareas();***

***if (listaTareas.empty()) {***

***return;***

***}***

***int indice;***

***cout << "Ingrese el número de la tarea a eliminar: ";***

***cin >> indice;***

***if (indice > 0 && indice <= listaTareas.size()) {***

***string tareaEliminada = listaTareas[indice - 1];***

***listaTareas.erase(listaTareas.begin() + (indice - 1));***

***cout << "Tarea '" << tareaEliminada << "' eliminada." << endl;***

***} else {***

***cout << "Número de tarea inválido." << endl;***

***}***

***} :*** Aca suceden varias cosas:

***. mostrarTareas();:*** Llama a la función ***mostrarTareas*** para mostrar las tareas disponibles antes de solicitar al usuario que seleccione una.

***. if (listaTareas.empty()) { return; }:*** Si la lista está vacía, termina la función porque no hay tareas que eliminar.

***. cin >> indice;:*** Pide al usuario el número de la tarea que quiere eliminar.

***. if (indice > 0 && indice <= listaTareas.size()):*** Verifica que el número ingresado sea válido. Si es así, elimina la tarea en la posición ***indice - 1*** usando ***erase.***

***void mostrarTareas() {***

***if (listaTareas.empty()) {***

***cout << "La lista de tareas está vacía." << endl;***

***} else {***

***cout << "\n--- Tareas ---" << endl;***

***for (size\_t i = 0; i < listaTareas.size(); ++i) {***

***cout << (i + 1) << ". " << listaTareas[i] << endl;***

***}***

***}***

***} :*** Aquí suceden varias cosas:

***. if (listaTareas.empty()):*** Si la lista está vacía, muestra un mensaje indicándolo.

***. for loop:*** Recorre la lista de tareas e imprime cada una junto a su número de posición.