ACTIVIDAD 3

PROGRAMACIÓN ESTRUCTURADA

CON ANSI C/C++

- *Profesor: York Mansilla*
- *Lenguaje: C/C++*
- *IDE: Visual Studio Code*
- *Compilador: MSYS2*

Brian Cruceño 5to 5ta

Proceso de instalacion Visual Studio Code

Primero instalar el Visual Studio Code por la página oficial, luego decidí configurarlo a gusto, para finalmente instalar el compilador MSYS2. Los links que utilicé son:

Brian Cruceño 5to 5ta

E.E.S.T. Nº 5
Gral. San Martín
Galileo Galilei

Gestor De Tareas


```cpp
#include <iostream>
#include <string>
#include <vector>

using namespace std;

int main() {
    vector<string> tareas;

    int opcion;
    do {
        cout << "1. Agregar tarea" << endl;
        cout << "2. Eliminar tarea" << endl;
        cout << "3. Mostrar tareas" << endl;
        cout << "4. Salir" << endl;

        cin >> opcion;

        switch (opcion) {
        case 1:
            cout << "Ingrese la tarea a registrar: ";
            string tarea;
            cin >> tarea;
            tareas.push_back(tarea);
            break;
        case 2:
            cout << "Ingrese la tarea a eliminar: ";
            string tareaEliminar;
            cin >> tareaEliminar;

            auto it = find(tareas.begin(), tareas.end(), tareaEliminar);
            if (it != tareas.end()) {
                tareas.erase(it);
            } else {
                cout << "Tarea no encontrada." << endl;
            }
            break;
        case 3:
            cout << "Lista de tareas:" << endl;
            for (const auto& tarea : tareas) {
                cout << tarea << endl;
            }
            break;
        }
    } while (opcion != 4);

    return 0;
}
```

Brian Cruceño 5to 5ta

E.E.S.T. Nº 5
Gral. San Martin
Galileo Galilei

Explicacion del codigo


```
1. Agregar tarea
2. Eliminar tarea
3. Mostrar tareas
4. Salir
1
Ingrese la tarea a registrar
Matematica
1. Agregar tarea
2. Eliminar tarea
3. Mostrar tareas
4. Salir
1
Ingrese la tarea a registrar
Lengua
1. Agregar tarea
2. Eliminar tarea
3. Mostrar tareas
4. Salir
2
Tarea a eliminar
1
1. Agregar tarea
2. Eliminar tarea
3. Mostrar tareas
4. Salir
3
Lista de tareas:
Lengua
1. Agregar tarea
2. Eliminar tarea
3. Mostrar tareas
4. Salir
```

#include <iostream>: Permite el uso de entrada y salida estándar, como cin y cout.

#include <string>: Proporciona soporte para la clase std::string, utilizada para manejar cadenas de texto.

#include <vector>: Incluye la clase std::vector, una colección dinámica que puede contener elementos como una lista

using namespace std;: Permite evitar el uso repetido del prefijo std:: antes de nombres como cin, cout, vector, etc.

int main(): Es el punto de entrada del programa donde se inicia la ejecución.

vector<string> tareas;: Declara un vector de cadenas llamado tareas para almacenar las tareas de la lista.

Brian Cruceño 5to 5ta

E.E.S.T. Nº 5
Gral. San Martín
Galileo Galilei

Explicacion del codigo

int opcion;: Declara una variable entera opcion para manejar las elecciones del usuario en el menú.

do {: Inicia un bucle do-while que asegura que el menú se ejecutará al menos una vez.

cout: Muestra el menú con las opciones disponibles para el usuario.

<< endl: Inserta un salto de línea al final de cada línea.

cin >> opcion;: Captura la elección del usuario y la almacena en la variable opcion.

switch (opcion): Evalúa el valor de opcion y ejecuta el código correspondiente al caso seleccionado.

case 1:: Inicia el bloque de código para la opción 1.

cout: Solicita al usuario que ingrese una tarea.

string tarea;: Declara una variable tarea para almacenar la tarea ingresada.

cin >> tarea;: Captura la tarea ingresada.

Brian Cruceño 5to 5ta

Explicacion del codigo

tareas.push_back(tarea);: Agrega la tarea al final del vector tareas.

break;: Finaliza la ejecución del caso actual.

case 2:: Inicia el bloque de código para la opción 2.

string tareaEliminar;: Declara una variable tareaEliminar para la tarea que se desea eliminar.

cin >> tareaEliminar;: Captura la tarea que se quiere eliminar.

auto it = find(...): Busca la tarea ingresada en el vector tareas. Devuelve un iterador que apunta al elemento encontrado o al final del vector si no se encuentra.

if (it != tareas.end()): Verifica si la tarea fue encontrada.

tareas.erase(it);: Si la tarea existe, elimina el elemento al que apunta el iterador it.

cout << "Tarea no encontrada.": Si no se encuentra la tarea, muestra un mensaje al usuario.

Explicacion del codigo

case 3:: Inicia el bloque de código para la opción 3.

for (const auto& tarea : tareas): Recorre cada tarea en el vector y la almacena en la variable tarea.

cout << tarea << endl;: Muestra cada tarea en una línea separada.

break;: Finaliza la ejecución del caso actual.

} while (opcion != 4);: Continúa mostrando el menú hasta que el usuario elija la opción 4 (Salir).

return 0;: Indica que el programa terminó exitosamente.

Brian Cruceño 5to 5ta

Final del informe.

Materia: Laboratorio de Programación
EEST N.o 5 - “Galileo Galilei” San Martín
5° año 5° jueves de 7:40 a 11:55 Hs

Brian Cruceño 5to 5ta