

Estructura del Informe

Análisis del Problema

1. Descripción del problema

Un colegio recién iniciado necesita actualizar las notas de sus estudiantes.

2. Requerimientos del sistema

- Funcionales

colas , listas , pilas y búsqueda

-

Evidencias de Trabajo en Equipo

1. Repositorio con Control de Versiones (Capturas de Pantalla)

```
#include <iostream>
#include <string>
using namespace std;

const int MAX_ESTUDIANTES = 15; //ponemos maximo del tamaño array

struct Estudiante { //definimos la estructura
    float nota;
    int id;
    string nombre;
};

void agregarEstudiante (Estudiante estudiantes[], int& contador, int id, float nota, string nombre) { //definimos funcion para agregar estudiantes
    for (int i = 0; i < contador; i++) {
        if (estudiantes[i].id == id) {
            cout<<"ESTE ESTUDIANTE YA ESTA REGISTRADO"<<endl;
            return;
        }
    }
    if (contador < MAX_ESTUDIANTES) {
        estudiantes[contador].id = id;
        estudiantes[contador].nota = nota;
        estudiantes[contador].nombre = nombre;
        contador++;
        cout<<"El estudiante con ID: "<<id<<" , nombre"<<nombre<<" , Notas = "<<nota<<endl;
    } else {
        cout<<"No se pueden agregar mas Estudiantes"<<endl;
    }
}

Estudiante* buscarEstudiante(Estudiante estudiantes[], int contador, int id) { //definimos la busqueda por puntero a un estudiante dentro del contador
    for (int i = 0; i < contador; i++){
        if (estudiantes[i].id == id) {
            return &estudiantes[i];
        }
    }
}

void actualizarNotaEstudiante(Estudiante& estudiante, float nuevaNota) { //definimos la funcion para actualizar un estudiante
    estudiante.nota = nuevaNota;
}

void mostrarEstudiantes(Estudiante estudiantes[], int contador) { //definimos la funcion para mostrar estudiantes
    for (int i = 0; i < contador; i++) {
        cout<<"Estudiante con ID: "<<estudiantes[i].id<<" , nombre: "<<estudiantes[i].nombre<<" , Notas: "<<estudiantes[i].nota<<endl;
    }
}

int main () { //ponemos el encabezado y sus variables tomando en cuenta el maximo que se permite en el array
    Estudiante estudiantes[MAX_ESTUDIANTES];
    int opcion;
    int contador = 0;
    float nota;
```

```

55 int buscarId;
56 string nombre;
57 int id;
58 do {
59     cout<<"\n---MENU PRINCIPAL---\n"<<endl;           //menu principal
60     cout<<"1.- Ingresar Estudiante"<<endl;
61     cout<<"2.- Buscar un Estudiante"<<endl;
62     cout<<"3.- Actualizar Notas"<<endl;
63     cout<<"4.- Mostrar total de estudiantes"<<endl;
64     cout<<"5.- Salir"<<endl;
65     cin>>opcion;
66
67     switch (opcion) {
68         case 1: {                                     //definimos la primera la opcion 1 para agregar un estudiante
69             cout<<"Ingrese ID del estudiante para agregar"<<endl;
70             cin>>id;
71             cout<<"Ingrese nota del estudiante"<<endl;
72             cin>>nota;
73             cout<<"Ingrese nombre del estudiante"<<endl;
74             cin.ignore();
75             getline(cin, nombre);
76             agregarEstudiante(estudiantes, contador, id, nota, nombre);
77             break;
78         }
79         case 2: {                                     //definimos la segunda opcion para buscar un estudiante
80             cout<<"Ingrese ID del estudiante para actualizar"<<endl;
81             cin>>buscarId;
82             Estudiante* est = buscarEstudiante(estudiantes, contador, buscarId);
83             if (est) {
84                 cout<<"Estudiante encontrado con ID: "<<est->id<<" , nombre: "<<est->nombre<<" , notas: "<<est->nota<<endl;
85             } else {
86                 cout<<"Estudiante con ID"<<buscarId<<" no encontrado"<<endl;
87             }
88             break;
89         }
90         case 3: {                                     //definimos la tercera opcion para actualizar una nota
91             cout<<"Ingrese ID del estudiante"<<endl;
92             cin>>buscarId;
93             Estudiante* est = buscarEstudiante(estudiantes, contador, buscarId);
94             if (est) {
95                 cout<<"Ingresar nueva nota"<<endl;
96                 float nuevaNota;
97                 cin>>nuevaNota;
98                 actualizarNotaEstudiante(*est, nuevaNota);
99                 cout<<"Notas de estudiante Actualizado"<<endl;
100             } else {
101                 cout<<"Estudiante"<<buscarId<<" no encontrado"<<endl;
102             }
103             break;
104         }
105         case 4: {                                     //definimos la cuarta opcion para mostrar todo el array y a todos los estudiantes
106             cout<<"Estudiantes actuales registrados"<<endl;
107             mostrarEstudiantes(estudiantes, contador);
108             break;

```

```

109     }
110     case 5: {
111         cout<<"Saliendo del programa"<<endl;
112         break;
113     }
114     default:
115         cout<<"Invalido, vuelva a intentarlo"<<endl;
116     }
117 }while (opcion !=5);
118 return 0;
119 }

```

https://github.com/ditmaredu/-GRUPO_E_Evaluacion01_Con2.git