

```

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

class Alumno {
private:
    string nombre;
    int edad;
    double nota1;
    double nota2;
    double nota3;
    string estado; // Aprobado o Desaprobado

public:
    // Constructor
    Alumno(string nombre, int edad, double nota1, double nota2, double nota3)
        : nombre(nombre), edad(edad), nota1(nota1), nota2(nota2), nota3(nota3) {
        actualizarEstado();
    }

    // Getters
    string getNombre() const { return nombre; }
    int getEdad() const { return edad; }
    double getNota1() const { return nota1; }
    double getNota2() const { return nota2; }
    double getNota3() const { return nota3; }
    string getEstado() const { return estado; }

    // Método para calcular el promedio
    double calcularPromedio() const {
        return (nota1 + nota2 + nota3) / 3;
    }

    // Método para determinar si un alumno está aprobado o no
    void actualizarEstado() {
        if (calcularPromedio() >= 7.0) {
            estado = "Aprobado";
        } else {
            estado = "Desaprobado";
        }
    }

    // Método para calcular la nota más alta
    double calcularNotaMaxima() const {
        return max({nota1, nota2, nota3});
    }
}

```

```

// Método para calcular la nota más baja
double calcularNotaMinima() const {
    return min({nota1, nota2, nota3});
}
};

// Método para buscar un alumno por su nombre
Alumno* buscarAlumno(vector<Alumno>& alumnos, const string& nombre) {
    for (auto& alumno : alumnos) {
        if (alumno.getNombre() == nombre) {
            return &alumno;
        }
    }
    return nullptr; // Retorna nullptr si no se encuentra el alumno
}

// Método para calcular el promedio general de todos los alumnos
double calcularPromedioGeneral(const vector<Alumno>& alumnos) {
    double sumaPromedios = 0.0;
    for (const auto& alumno : alumnos) {
        sumaPromedios += alumno.calcularPromedio();
    }
    return sumaPromedios / alumnos.size();
}

int main() {
    // Vector para almacenar los alumnos
    vector<Alumno> alumnos;

    // Ingresar datos de los alumnos y agregarlos al vector
    string nombre;
    int edad;
    double nota1, nota2, nota3;
    int numAlumnos;

    cout << "Ingrese el número de alumnos: ";
    cin >> numAlumnos;

    for (int i = 0; i < numAlumnos; i++) {
        cout << "Ingrese el nombre del alumno: ";
        cin >> nombre;
        cout << "Ingrese la edad del alumno: ";
        cin >> edad;
        cout << "Ingrese la nota 1 del alumno: ";
        cin >> nota1;
        cout << "Ingrese la nota 2 del alumno: ";
        cin >> nota2;
        cout << "Ingrese la nota 3 del alumno: ";
    }
}

```

```

        cin >> nota3;

        alumnos.emplace_back(nombre, edad, nota1, nota2, nota3);
    }

    // Mostrar los datos de los alumnos y sus promedios
    cout << "\nDatos de los alumnos:" << endl;
    for (const auto& alumno : alumnos) {
        cout << "Nombre: " << alumno.getNombre() << endl;
        cout << "Edad: " << alumno.getEdad() << endl;
        cout << "Nota 1: " << alumno.getNota1() << endl;
        cout << "Nota 2: " << alumno.getNota2() << endl;
        cout << "Nota 3: " << alumno.getNota3() << endl;
        cout << "Promedio: " << alumno.calcularPromedio() << endl;
        cout << "Estado: " << alumno.getEstado() << endl;
        cout << "Nota más alta: " << alumno.calcularNotaMaxima() << endl;
        cout << "Nota más baja: " << alumno.calcularNotaMinima() << endl;
        cout << endl;
    }

    // Búsqueda de un alumno por nombre
    string nombreBusqueda;
    cout << "Ingrese el nombre del alumno a buscar: ";
    cin >> nombreBusqueda;
    Alumno* alumnoEncontrado = buscarAlumno(alumnos, nombreBusqueda);

    if (alumnoEncontrado) {
        cout << "Alumno encontrado:" << endl;
        cout << "Nombre: " << alumnoEncontrado->getNombre() << endl;
        cout << "Edad: " << alumnoEncontrado->getEdad() << endl;
        cout << "Promedio: " << alumnoEncontrado->calcularPromedio() << endl;
        cout << "Estado: " << alumnoEncontrado->getEstado() << endl;
    } else {
        cout << "Alumno no encontrado." << endl;
    }

    // Calcular y mostrar el promedio general de todos los alumnos
    double promedioGeneral = calcularPromedioGeneral(alumnos);
    cout << "\nPromedio general de todos los alumnos: " << promedioGeneral << endl;

    return 0;
}

```