Actividad 22

Docente: Jimmy Nataniel Requena Llorentty

Materia: Programación III

Carrera: Ingeniería En Sistemas

Estudiantes: Joaquin Marcos Maita Flores

Santa Cruz – Bolivia

2025

Titulo: Actividad 22

Estudiante/s: Joaquin Marcos Maita Flores

Encapsulando al Estudiante con error

```
#include <iostream>
#include <string>
#include <stdexcept> // Para manejo de excepciones (opcional)

class Estudiante {
private:
    std::string nombre;
    int edad;
    std::string matricula;
    double promedio;
    std::string carrera; // NUEVO atributo
    std::string correo;  // NUEVO atributo

public:
    // Constructor: se ejecuta al crear un nuevo objeto Estudiante
    // Permite asignar valores iniciales al objeto
    Estudiante(std::string nom, int ed, std::string matr, std::string carr
= "", std::string mail = "") {
        setNombre(nom);
        setEdad(ed);
        matricula = matr;
        promedio = 0.0;
        setCarrera(carr);
        setCorreo(mail);
        std::cout << "Estudiante '" << nombre << "' creado." << std::endl;
    }

    // Getters
    // Métodos para obtener los valores privados desde fuera
    std::string getNombre() const { return nombre; }
    int getEdad() const { return edad; }
    std::string getMatricula() const { return matricula; }
    double getPromedio() const { return promedio; }
    std::string getCarrera() const { return carrera; }
    std::string getCorreo() const { return correo; }

    // Setters con validación
    // Métodos para modificar valores privados de forma segura (con
validaciones)
    void setNombre(const std::string& nuevoNombre) {
        if (!nuevoNombre.empty()) {
            nombre = nuevoNombre;
        } else {
            std::cout << "Error: El nombre no puede estar vacío." <<
std::endl;
        }
```

Carrera: Ingeniería En Sistemas

Materia: Programación III

Titulo: Actividad 22

Estudiante/s: Joaquin Marcos Maita Flores

```
    }

    void setEdad(int nuevaEdad) {
        if (nuevaEdad >= 5 && nuevaEdad <= 100) {// Rango válido de edad
            edad = nuevaEdad;
        } else {
            std::cout << "Error: Edad '" << nuevaEdad << "' inválida para
el estudiante " << nombre << "." << std::endl;
        }
    }

    void setPromedio(double nuevoPromedio) {
        if (nuevoPromedio >= 0.0 && nuevoPromedio <= 10.0) {
            promedio = nuevoPromedio;
        } else {
            std::cout << "Error: Promedio '" << nuevoPromedio << "'
inválido para " << nombre << "." << std::endl;
        }
    }

    void setCarrera(const std::string& nuevaCarrera) {
        if (!nuevaCarrera.empty()) {
            carrera = nuevaCarrera;
        } else {
            carrera = "No especificada";
        }
    }

    void setCorreo(const std::string& nuevoCorreo) {
        // Validamos que contenga al menos un '@'
        if (nuevoCorreo.find('@') != std::string::npos) {
            correo = nuevoCorreo;
        } else {
            std::cout << "Error: Correo electrónico inválido." <<
std::endl;
        }
    }
      // Método para mostrar todos los datos del estudiante

    void mostrarInformacion() const {
        std::cout << "------------------------------" << std::endl;
        std::cout << "Nombre: " << nombre << std::endl;
        std::cout << "Edad: " << edad << " años" << std::endl;
        std::cout << "Matrícula: " << matricula << std::endl;
        std::cout << "Promedio: " << promedio << std::endl;
        std::cout << "Carrera: " << carrera << std::endl;
        std::cout << "Correo: " << correo << std::endl;
        std::cout << "------------------------------" << std::endl;
    }
```

Carrera: Ingeniería En Sistemas

Materia: Programación III

Titulo: Actividad 22

Estudiante/s: Joaquin Marcos Maita Flores

```
};

int main() {
    Estudiante estudiante1("Juaquin Soliz", 20, "A123", "Ingeniería",
"juaquin@upds.edu");
    estudiante1.mostrarInformacion();

    std::cout << "\nIntentando actualizar edad, promedio y correo..." <<
std::endl;
    estudiante1.setEdad(21); // Edad válida
    estudiante1.setPromedio(8.5); // Promedio válido
    estudiante1.setCorreo("correo-invalido"); // Prueba de validación
(falta '@')
    estudiante1.setEdad(150); // Edad inválida (fuera de rango)
    estudiante1.setPromedio(-2.0); // Promedio inválido

    std::cout << "\nInformación actualizada de " <<
estudiante1.getNombre() << ":" << std::endl;
    estudiante1.mostrarInformacion();

     //Descomentar estas líneas para observar los errores de compilador
por acceso indebido:
     //estudiante1.edad = 25; // ERROR: 'edad' es privado
     //std::cout << estudiante1.promedio; // ERROR: 'promedio' también es
privado

    // Cómo observar los errores:
    std::cout << "\nSi descomentas las líneas anteriores y compilas, verás
errores como:" << std::endl;
    std::cout << "'int Estudiante::edad' is private within this context"
<< std::endl;
    std::cout << "Esto enseña a respetar el encapsulamiento y a usar
setters/getters." << std::endl;

    // Estudiante con datos inválidos al inicio
    Estudiante estudiante2("Priscila Vaca", -10, "B456", "", "pvaca.com");
    estudiante2.mostrarInformacion();
    std::cout << "\033[33mJoaquin Marcos Maita Flores.\033[0m" <<
std::endl;
    std::cout << "\033[33mSIN ERRORES.\033[0m" << std::endl;

    return 0;
}
```

Carrera: Ingeniería En Sistemas

Materia: Programación III

Titulo: Actividad 22

Estudiante/s: Joaquin Marcos Maita Flores

Corriendo Código CON ERROR

```
Compilation failed due to following error(s).
main.cpp: In function 'int main()':
main.cpp:107:17: error: 'int Estudiante::edad' is private within this context
  107 |     estudiante1.edad = 25; // ERROR: 'edad' es privado
      |                 ^~~~
main.cpp:8:9: note: declared private here
    8 |     int edad;
      |         ^~~~
main.cpp:107:17: note: field 'int Estudiante::edad' can be accessed via 'int Estudiante::getEdad() const'
  107 |     estudiante1.edad = 25; // ERROR: 'edad' es privado
      |                 ^~~~
      |                 getEdad()
main.cpp:108:31: error: 'double Estudiante::promedio' is private within this context
  108 |     std::cout << estudiante1.promedio; // ERROR: 'promedio' también es privado
      |                               ^~~~~~~~
main.cpp:10:12: note: declared private here
   10 |     double promedio;
      |            ^~~~~~~~
main.cpp:108:31: note: field 'double Estudiante::promedio' can be accessed via 'double Estudiante::getPromedio() const'
  108 |     std::cout << estudiante1.promedio; // ERROR: 'promedio' también es privado
      |                               ^~~~~~~~
      |                               getPromedio()
```

```
    //Descomentar estas líneas para observar los errores de compilador por acceso indebido:
    estudiante1.edad = 25; // ERROR: 'edad' es privado
    std::cout << estudiante1.promedio; // ERROR: 'promedio' también es privado

    // Cómo observar los errores:
    std::cout << "\nSi descomentas las líneas anteriores y compilas, verás errores como:" << std::endl;
    std::cout << "'int Estudiante::edad' is private within this context" << std::endl;
    std::cout << "Esto enseña a respetar el encapsulamiento y a usar setters/getters." << std::endl;

    // Estudiante con datos inválidos al inicio
    Estudiante estudiante2("Priscila Vaca", -10, "B456", "", "pvaca.com");
    estudiante2.mostrarInformacion();
    std::cout << "\033[33mJoaquin Marcos Maita Flores.\033[0m" << std::endl;
    std::cout << "\033[33mSIN ERRORES.\033[0m" << std::endl;

    return 0;
}
```

Estas líneas generarían errores si se descomentan porque intentan acceder a atributos privados

// estudiante1.edad = 25; // ERROR: edad es privado

// std::cout << estudiante1.promedio; // ERROR: promedio es privado

Carrera: Ingeniería En Sistemas

Materia: Programación III

Titulo: Actividad 22

Estudiante/s: Joaquin Marcos Maita Flores

Corriendo Código SIN ERROR

¿Qué aprendimos?

- Cómo encapsular correctamente usando atributos privados.
- Por qué se usan getters y setters con validación.
- Cómo evitar errores y proteger la lógica del programa.
- Qué ocurre cuando intentas acceder directamente a variables privadas.

Carrera: Ingeniería En Sistemas

Materia: Programación III

Titulo: Actividad 22

Estudiante/s: Joaquin Marcos Maita Flores

- Cómo validar entradas como correo, edad o promedio.
- responsabilidades.

anexo

https://onlinegdb.com/hcu7VNU3i

Fecha y hora actual: 2025-06-17 21:53:16

Carrera: Ingeniería En Sistemas

Materia: Programación III