## **Nome: Erison Oliveira de Sousa**

## 

## **Sistema Acadêmico em C++**

### **Objetivo do Projeto**

Esse projeto tem como objetivo mostrar na prática o uso da Programação Orientada a Objetos (POO) em C++. A ideia foi criar um sistema que representa diferentes tipos de usuários de um ambiente acadêmico, como alunos, professores, monitores e funcionários.

Com isso, foi possível aplicar:

* Herança e herança múltipla
* Polimorfismo
* Sobrecarga e sobrescrita de métodos
* Organização do código com vários arquivos .h e .cpp

### **Organização dos Arquivos**

* main.cpp: onde o programa principal é executado
* usuario.h/.cpp: classe base para todos os usuários
* aluno.h/.cpp: representa os alunos
* professor.h/.cpp: representa os professores
* funcionarioadministrativo.h/.cpp: representa os funcionários
* monitor.h/.cpp: classe que herda de aluno e professor (herança múltipla)

### **Estrutura das Classes**

#### **Classe usuario (classe base)**

* Atributos: nome, email, tipo
* Método virtual: gerarRelatorio() — precisa ser implementado nas classes filhas

#### **Classe aluno**

* Herda de usuario
* Atributos: matricula, curso, disciplinas
* Métodos: adicionarDisciplina() e gerarRelatorio() (tem sobrecarga com opção de mostrar notas)

#### **Classe professor**

* Herda de usuario
* Atributos: área de atuação, disciplinas ministradas
* Método: gerarRelatorio()

#### **Classe funcionarioadministrativo**

* Herda de usuario
* Atributos: departamento, cargo
* Método: gerarRelatorio()

#### **Classe monitor**

* Herda de aluno e professor
* Atributo: disciplinas monitoradas
* Métodos: adicionarDisciplinaMonitorada() e gerarRelatorio()
* Também possui um método para redirecionar adicionarDisciplina() corretamente (por causa da herança múltipla)

### **Execução do Programa**

No main.cpp, são criados objetos de cada tipo de usuário e armazenados em um vetor de ponteiros do tipo usuario\*. Com isso, o polimorfismo permite que o programa chame o gerarRelatorio() correto para cada tipo de objeto.

### **Como Compilar**

Para compilar, basta usar este comando no terminal:

bash

CopiarEditar

g++ main.cpp usuario.cpp aluno.cpp professor.cpp funcionarioadministrativo.cpp monitor.cpp -o sistema

Depois de compilar, execute com:

* Windows: sistema.exe
* Linux/macOS: ./sistema

### **Conclusão**

O projeto foi importante para entender como funciona a herança em C++, incluindo herança múltipla. Também ajudou a praticar o uso de métodos virtuais, sobrescrita, sobrecarga e organização do código em arquivos separados.