

**CENTRO DE EDUCAÇÃO TECNOLÓGICA DO AMAZONAS – CETAM
ESCOLA ESTADUAL DE TEMPO INTEGRAL GOVERNADOR MELO E PÓVOAS
CURSO TÉCNICO DE NÍVEL MÉDIO EM INFORMÁTICA**

**DESENVOLVIMENTO DE UM SISTEMA ESCOLAR UTILIZANDO
PROGRAMAÇÃO ORIENTADA A OBJETOS
Modelagem, implementação e testes de um sistema modular em C++**

**MANAUS – AM
2025**

**FRANCINEI BATISTA TAVARES
ARGENIS JOSUE FUENTES OSORIO
OTAVIO LUIZ TAVERA DO NASCIMENTO
JOHN SAYMON FURTADO MELO
EDMA STEFANIA LOPEZ ZORRILLA
YASMIN GAMA PARA**

Relatório técnico atividades práticas, apresentado
como
requisito parcial para aprovação na unidade
curricular/curso: linguagem de programação I –
Técnico
de Nível Médio em Informática.

Professora Esp. Priscila Gonçalves.

**MANAUS – AM
2025**

SUMÁRIO

1. Introdução	4
2. Objetivo Geral	4
3. Objetivos Específicos	4
4. Desenvolvimento – Exemplos Práticos em POO	5
4.1 Classe Base – Pessoa	5
4.2 Classe Derivada – Aluno	5
4.3 Classe Secretaria	5
4.4 Polimorfismo	6
4.5 Teste de Integração	6
5. Metodologia Utilizada	6
5.1 Levantamento de Requisitos	6
5.2 Modelagem UML	7
5.3 Programação Orientada a Objetos	7
5.4 Programação em Pares	7
5.5 Testes e Refatoração	7
6. Conclusão	7
7. Referência Bibliográfica	8

1. Introdução

O avanço das tecnologias de informação tem impulsionado instituições de ensino a modernizarem seus processos internos e administrativos. Nesse contexto, os sistemas escolares tornam-se ferramentas essenciais para organizar dados de alunos, setores, funcionários e atividades da secretaria. O desenvolvimento de um sistema desse tipo utilizando Programação Orientada a Objetos (POO) possibilita uma estrutura modular, reutilizável e de fácil manutenção.

O presente relatório descreve o processo de construção de um Sistema Escolar em C++, abordando desde a definição do problema até a implementação, testes, metodologia aplicada e resultados obtidos. Além disso, exemplifica na prática como os conceitos fundamentais de POO — encapsulamento, herança, polimorfismo e integração entre classes — foram utilizados para representar de forma fiel os componentes de uma instituição escolar.

2. Objetivo Geral

Desenvolver um sistema escolar modular e orientado a objetos em C++, aplicando técnicas de modelagem, encapsulamento, herança, polimorfismo e integração entre classes para representar e gerenciar as operações administrativas de uma instituição de ensino.

3. Objetivos Específicos

- Estruturar o sistema escolar por meio de classes e objetos representativos.
- Aplicar o encapsulamento para proteger e organizar atributos internos.
- Implementar classes derivadas utilizando herança.
- Demonstrar o uso de polimorfismo em métodos sobrescritos.
- Realizar testes de integração entre classes.
- Refatorar o código para melhorar legibilidade e eficiência.
- Documentar todas as etapas conforme normas técnicas e acadêmicas.

4. Desenvolvimento – Exemplos Práticos em POO

A seguir, trechos que exemplificam a aplicação prática da POO no sistema escolar.

4.1 Classe Base – Pessoa

```
class Pessoa {  
protected:  
    string nome;  
    int idade;  
  
public:  
    Pessoa(string nome, int idade) : nome(nome), idade(idade) {}  
  
    virtual void exibirInfo() {  
        cout << "Nome: " << nome << ", Idade: " << idade << endl;  
    }  
};
```

4.2 Classe Derivada – Aluno

```
class Aluno : public Pessoa {  
private:  
    string matricula;  
  
public:  
    Aluno(string nome, int idade, string matricula)  
        : Pessoa(nome, idade), matricula(matricula) {}  
  
    void exibirInfo() override {  
        cout << "[Aluno] Nome: " << nome  
            << ", Idade: " << idade  
            << ", Matrícula: " << matricula << endl;  
    }  
};
```

4.3 Classe Secretaria

```
class Secretaria {  
private:  
    vector<Aluno> alunos;
```

```

public:
    void cadastrarAluno(const Aluno &aluno) {
        alunos.push_back(aluno);
    }

    void listarAlunos() {
        for (const auto &a : alunos) {
            a.exibirInfo();
        }
    }
};

```

4.4 Polimorfismo

```

void mostrarPessoa(Pessoa* p) {
    p->exibirInfo(); // comportamento depende da classe associada
}

```

4.5 Teste de Integração

```

int main() {
    Secretaria secretaria;
    Aluno aluno1("Ana", 16, "MAT001");
    Aluno aluno2("Lucas", 17, "MAT002");

    secretaria.cadastrarAluno(aluno1);
    secretaria.cadastrarAluno(aluno2);

    secretaria.listarAlunos();

    return 0;
}

```

5. Metodologia Utilizada

O desenvolvimento do sistema seguiu uma metodologia estruturada contendo:

5.1 Levantamento de Requisitos

Identificação das necessidades do sistema e definição dos requisitos funcionais e não funcionais.

5.2 Modelagem UML

Criação de um diagrama de classes para representar os relacionamentos entre Pessoa, Aluno, Secretaria e Setores.

5.3 Programação Orientada a Objetos

Os códigos foram desenvolvidos aplicando rigorosamente:

- Encapsulamento
- Herança
- Polimorfismo
- Modularização
- Separação de classes em arquivos .h e .cpp

5.4 Programação em Pares

As etapas de implementação foram realizadas em dupla, permitindo revisão constante e divisão eficiente das responsabilidades.

5.5 Testes e Refatoração

Testes de integração entre classes, seguidos de melhorias no código para garantir clareza, manutenção e eficiência.

6. Conclusão

O desenvolvimento do Sistema Escolar em C++ permitiu aplicar e consolidar os conceitos essenciais de Programação Orientada a Objetos em um projeto prático e funcional. O uso de herança, encapsulamento, polimorfismo e modelagem estruturada possibilitou criar uma arquitetura robusta e extensível. A metodologia empregada e o trabalho colaborativo contribuíram para aprimorar tanto o entendimento teórico quanto a prática de desenvolvimento em equipes. O sistema final atende aos requisitos estabelecidos e demonstra o potencial da POO na construção de soluções organizadas e escaláveis.

7. Referência Bibliográfica

- ABNT. NBR 10719: Informação e documentação – Relatórios técnicos e/ou científicos. 2015.
- DEITEL, P.; DEITEL, H. *C++ Como Programar*. São Paulo: Pearson.
- SOMMERVILLE, Ian. *Engenharia de Software*. Pearson.
- STROUSTRUP, Bjarne. *The C++ Programming Language*. Addison-Wesley.
- GAMMA, E. et al. *Padrões de Projeto*. Bookman.