



A respeito da implementação de aplicações orientada a objetos utilizando a linguagem de programação C++, qual a diferença entre:

1. **Métodos virtuais puros;**
2. **Classes abstratas;**
3. **Interfaces.**

1. Métodos Virtuais puros são assinaturas feitas em classes Abstratas que não possuem lógica implementada, deixando para a classe Derivada a implementação da lógica.

```
class Base{
public:
    void fooVirtual(){ cout << "Esse não é um método virtual";};
    void fooVirtualPuro() = 0; //Essa função não tem definição lógica
                             //portanto deve ser reimplementada na deriv
};

class Derivada : public Base{
public:
    void fooVirtualPuro(){ cout << "Essa função é um método virtual puro";};
};

int main(){
    Derivada a;
    a.fooVirtualPuro();
}
```

2. Classes Abstratas são classes que possuem pelo menos um métodos virtual puro em sua implementação, servindo como modelo para as classes Derivadas.

```
class Funcionario{
public:
    ~Funcionario();
    virtual void trabalhar() = 0;
    float salario;
};

class Administrador : public Funcionario{
public:
    void trabalhar(){ std::cout << "Eu sonego imposto e recebo R$" <<
                        salario << " por isso" << std::end;};
};

int main(){
    Funcionario* a = new Administrador();
    a->salario = 1200;
    a->trabalhar();
}
```



3. Interfaces (apesar de não existirem em C++), se assemelham às classes Abstratas, com a principal diferença que TODOS os métodos são virtuais puros. Classes Derivadas podem implementar mais de uma interface.

```
class SerVivo{
public:
    ~SerVivo();
    virtual void viver() = 0;
};

class Estudante{
public:
    virtual ~Estudante();
    virtual void estudar();
};

class Crianca : public SerVivo, public Estudante {
public:
    void viver() { std::cout << "Eu vivo pra comer reboco e brincar com
                        besouros" << std::endl; };
    void estudar() { std::cout << "Eu vou pra escola para comer massinha
                        e cola" << std::endl; };
};

class Adulto : public SerVivo, public Estudante {
public:
    void viver() { std::cout << "Eu vivo ansioso e com medo de não
                        conseguir pagar minhas contas" <<
                        std::endl; };
    void estudar() { std::cout << "Eu pago um absurdo para fazer faculdade
                        e reprovo mesmo assim" << std::endl; };
};

int main(){

    Crianca a;
    Adulto b;

    a.viver();
    a.estudar();

    b.viver();
    b.estudar();

}
```