

# A ESSÊNCIA DA ORIENTAÇÃO A OBJETOS



**JOSÉ CARLOS ESTEVES**

# POO

NA VASTA PAISAGEM DA PROGRAMAÇÃO, A PROGRAMAÇÃO ORIENTADA A OBJETOS (POO) DESTACA-SE COMO UMA ABORDAGEM FUNDAMENTAL. SEUS QUATRO PILARES - ABSTRAÇÃO, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO - CONSTITUEM A ESPINHA DORSAL DESSA METODOLOGIA. VAMOS EXPLORAR CADA UM DELES COM EXEMPLOS SIMPLES E PRÁTICOS.

1

# ABSTRAÇÃO

---

Simplificando o Complexo

# ABSTRAÇÃO

A ABSTRAÇÃO PERMITE QUE NOS CONCENTREMOS NOS ASPECTOS ESSENCIAIS DE UM OBJETO, IGNORANDO DETALHES MENOS RELEVANTES. IMAGINE UM CARRO. PARA UM MOTORISTA, O QUE IMPORTA SÃO OS CONTROLES: VOLANTE, PEDAIS E ALAVANCA DE CÂMBIO. A COMPLEXIDADE DO MOTOR E DO SISTEMA DE TRANSMISSÃO É ABSTRAÍDA, PERMITINDO QUE O MOTORISTA SE CONCENTRE NA TAREFA DE DIRIGIR.

```
code.cs - José Carlos Esteves

class Carro
{
    public string Marca { get; set; }
    public string Modelo { get; set; }

    public void Dirigir()
    {
        Console.WriteLine($"Dirigindo o {Modelo}");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Carro meuCarro = new Carro();
        meuCarro.Marca = "Toyota";
        meuCarro.Modelo = "Corolla";
        meuCarro.Dirigir();
    }
}
```

2

# ENCAPSULAMENTO

---

Protegendo o interior

# ENCAPSULAMENTO

O ENCAPSULAMENTO ESCONDE A IMPLEMENTAÇÃO INTERNA DE UM OBJETO E EXPÕE APENAS A INTERFACE PÚBLICA NECESSÁRIA PARA INTERAGIR COM ELE. VOLTANDO AO EXEMPLO DO CARRO, UM MOTORISTA NÃO PRECISA ENTENDER A MECÂNICA INTERNA DO VEÍCULO PARA CONDUZI-LO COM SEGURANÇA. AO OCULTAR AS PARTES INTERNAS, OS DADOS FICAM PROTEGIDOS DE ALTERAÇÕES INDESEJADAS.

```
code.cs - José Carlos Esteves

class ContaBancaria
{
    public string Titular { get; set; }
    private decimal Saldo { get; set; }

    public void Depositar(decimal valor)
    {
        Saldo += valor;
    }

    public void Sacar(decimal valor)
    {
        if (Saldo >= valor)
        {
            Saldo -= valor;
        }
        else
        {
            Console.WriteLine("Saldo insuficiente");
        }
    }

    public decimal ConsultarSaldo()
    {
        return Saldo;
    }
}
```

3

# HERANÇA

---

Construindo sobre fundações sólidas

# HERANÇA

A HERANÇA É UM CONCEITO QUE PERMITE A CRIAÇÃO DE HIERARQUIAS DE CLASSES, ONDE UMA CLASSE MAIS ESPECÍFICA PODE HERDAR OS ATRIBUTOS E MÉTODOS DE UMA CLASSE MAIS GENÉRICA. ISSO PERMITE A REUTILIZAÇÃO DE CÓDIGO, POIS CARACTERÍSTICAS COMUNS PODEM SER DEFINIDAS EM UMA CLASSE BASE, E CLASSES DERIVADAS PODEM ESTENDER OU PERSONALIZAR ESSE COMPORTAMENTO. ESSE PILAR CONTRIBUI PARA A ECONOMIA DE TEMPO NA CODIFICAÇÃO, EVITANDO DUPLICAÇÃO DE CÓDIGO.

```
code.cs - José Carlos Esteves

class Veiculo
{
    public string Marca { get; set; }
    public string Modelo { get; set; }

    public void Dirigir()
    {
        Console.WriteLine($"Dirigindo um {Marca}
{Modelo}");
    }
}

class Carro : Veiculo
{
    public int NumeroPortas { get; set; }

    // Construtor específico para carro
    public Carro(string marca, string modelo, int
numeroPortas)
    {
        Marca = marca;
        Modelo = modelo;
        NumeroPortas = numeroPortas;
    }
}
```



4

# POLIMORFISMO

---

Múltiplas formas de ser

# POLIMORFISMO

O POLIMORFISMO PERMITE QUE OBJETOS DE DIFERENTES CLASSES SEJAM TRATADOS DE MANEIRA UNIFORME, RESPONDENDO DE FORMA DIFERENTE A MÉTODOS COM O MESMO NOME. ISSO AMPLIA A FLEXIBILIDADE E A EXPRESSIVIDADE DO CÓDIGO.

```
code.cs - José Carlos Esteves

class Veiculo
{
    public virtual void Dirigir()
    {
        Console.WriteLine("Dirigindo um veículo genérico");
    }
}

class Carro : Veiculo
{
    public override void Dirigir()
    {
        Console.WriteLine("Dirigindo um carro");
    }
}

class Caminhao : Veiculo
{
    public override void Dirigir()
    {
        Console.WriteLine("Dirigindo um caminhão");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Veiculo veiculo1 = new Carro();
        Veiculo veiculo2 = new Caminhao();

        DirigirVeiculo(veiculo1);
        DirigirVeiculo(veiculo2);
    }

    static void DirigirVeiculo(Veiculo veiculo)
    {
        veiculo.Dirigir();
    }
}
```

# OBRIGADO POR LER

ESTE E-BOOK TEVE A IMAGEM DA CAPA GERADA POR IA E SEU CONTEÚDO DIAGRAMADO E REDIGIDO POR JOSÉ CARLOS ESTEVES.



[www.linkedin.com/JoseEsteves](https://www.linkedin.com/JoseEsteves)



[github.com/JoseCarlos67](https://github.com/JoseCarlos67)