

Padrão de Projeto Abstract Factory

Aplicação em um sistema de montagem de veículos

Autores: Thiago Henrique, João Paulo, João Pedro

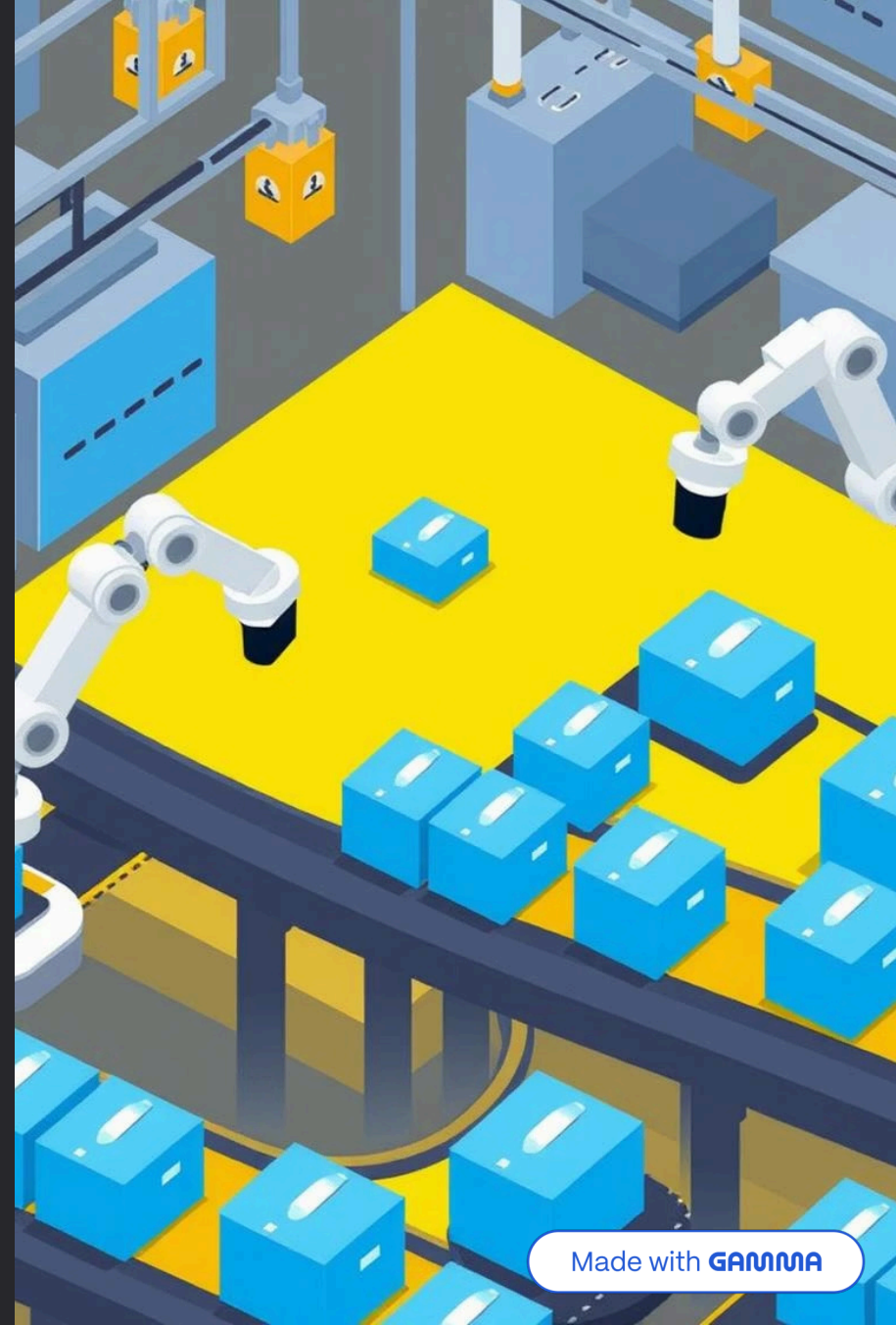
Curso: Ciência da Computação

Universidade: Unifenas



O que é Abstract Factory?

- 1 Padrão Criacional GoF
Catálogo de padrões de projeto Gang of Four.
- 2 Famílias de Objetos
Cria objetos relacionados sem acoplamento direto.
- 3 Múltiplas Variantes
Ideal para conjuntos de produtos com variações.



Problema e Motivação

Cenário de Montagem

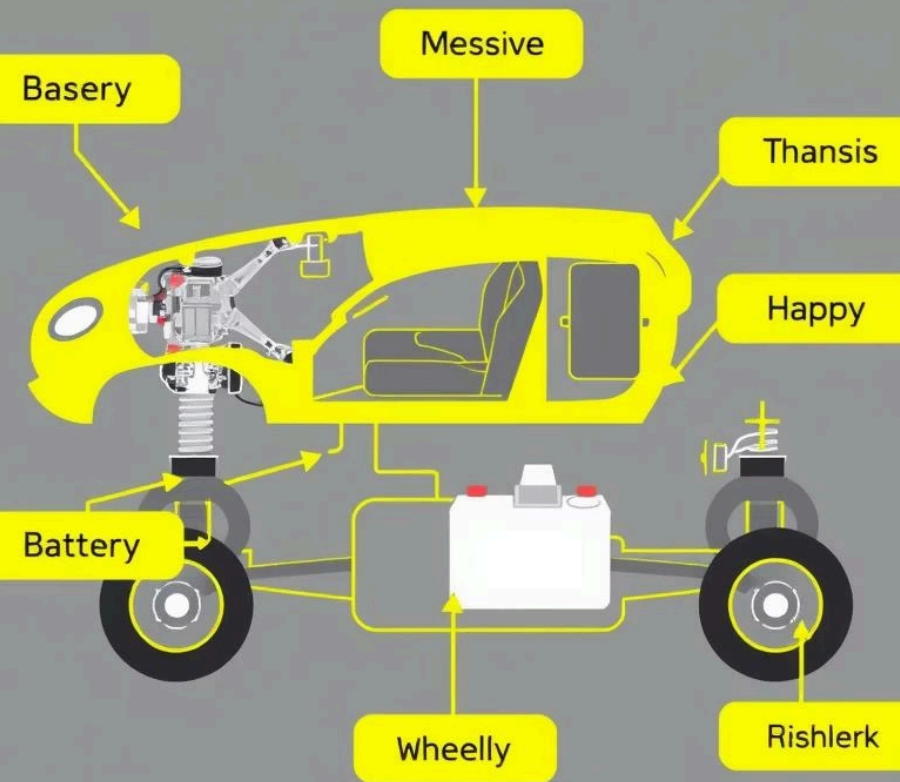
- Montadora virtual de veículos.
- Veículos a combustão ou elétricos.

Desafio de Compatibilidade

- Garantir componentes compatíveis (motor, tanque, bateria).
- Evitar incompatibilidades entre tipos de veículos.

Solução com Abstract Factory

- Criação correta e flexível de objetos.
- Assegura a integridade do veículo montado.



Estrutura do Padrão



VeiculoFactory

Interface abstrata
para criar
componentes.



Fábricas Concretas

CombustaoFactory,
EletricoFactory.



Produtos

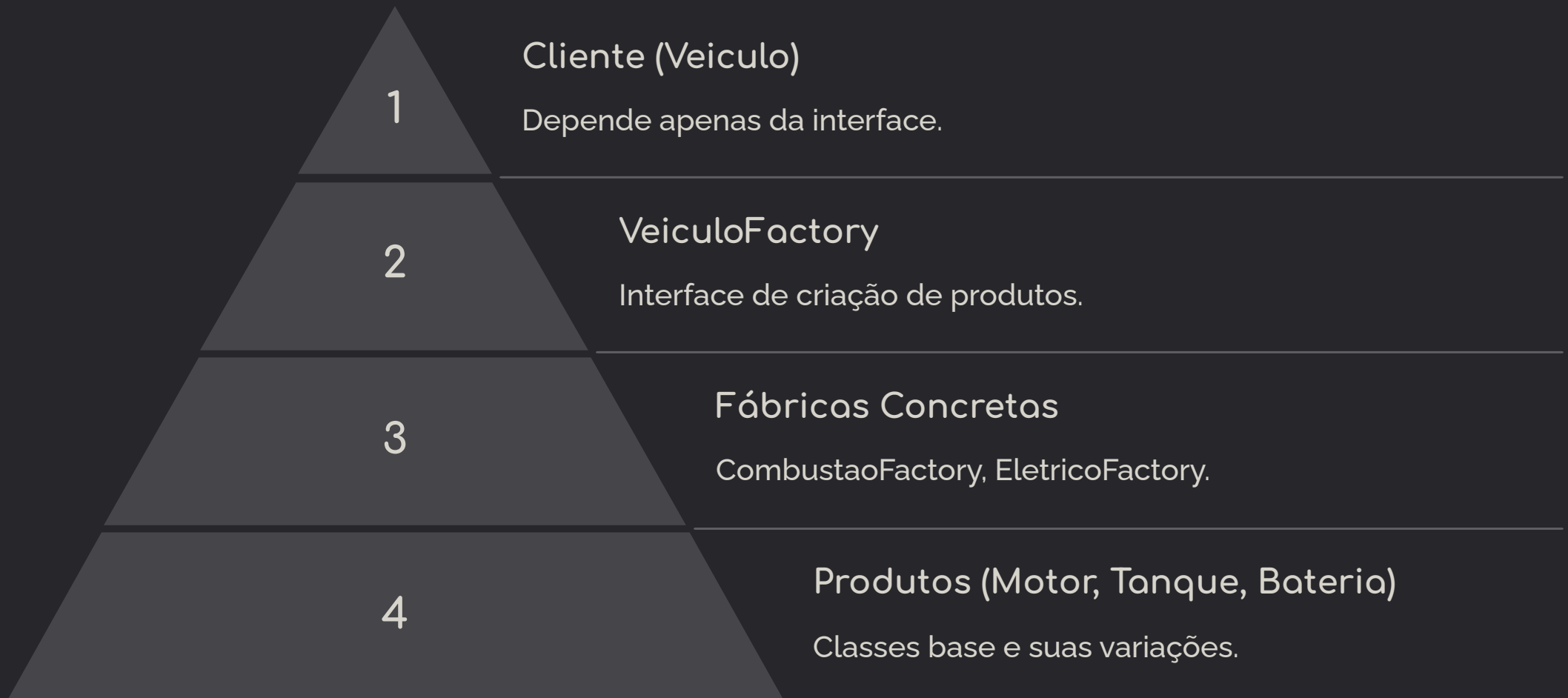
Motor, Tanque,
Bateria (variantes).



Cliente

Classe Veiculo, usa
a fábrica.

Diagrama de Classes UML



Código - Interface e Fábricas

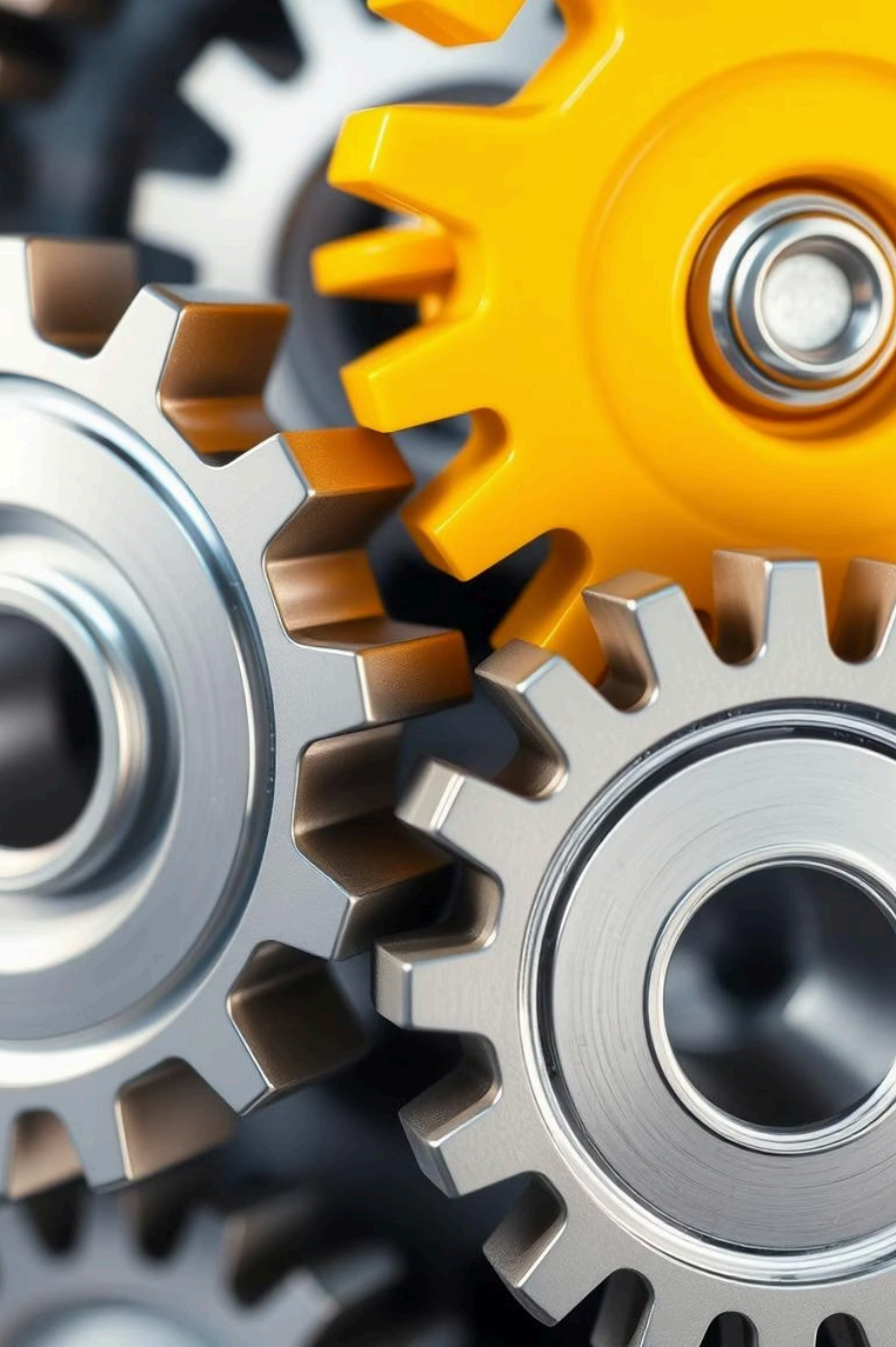
```
public interface VeiculoFactory {  
    Motor criarMotor();  
    Tanque criarTanque();  
    Bateria criarBateria();  
}
```

```
public class EletricoFactory implements VeiculoFactory {  
    public Motor criarMotor() {  
        return new MotorEletrico();  
    }  
    public Tanque criarTanque() {  
        return null;  
    }  
    public Bateria criarBateria() {  
        return new Bateria300km();  
    }  
}
```



Código - Classe Cliente

```
public class Veiculo {  
    private Motor motor;  
    private Tanque tanque;  
    private Bateria bateria;  
  
    public Veiculo(VeiculoFactory factory) {  
        motor = factory.criarMotor();  
        tanque = factory.criarTanque();  
        bateria = factory.criarBateria();  
    }  
  
    public void mostrarInfo() {  
        if (motor != null) motor.infoMotor();  
        if (tanque != null) tanque.infoTanque();  
        if (bateria != null) bateria.infoBateria();  
    }  
}
```



Benefícios do Padrão



Modularidade e Manutenção

Código organizado e fácil