

# Factory

Padrão de Projeto de Criação

**INTEGRANTES:** Arthur Monteiro, Brenda Stefany, Bruna Letícia, Elenice Florentina,  
Eduarda Faria e Marco Aurélio de Faria

# FACTORY

- O Factory Method é um padrão de criação (creational pattern) que define uma interface para criar objetos, mas permite que as subclasses decidam qual classe concreta instanciar.
- Um dos padrões mais utilizados no desenvolvimento de software, especialmente em programação orientada a objetos.
- Fornece uma maneira de criar objetos sem especificar a classe exata do objeto que será criado.

# UTILIZAÇÃO

O código precisa decidir qual objeto criar, usando vários if/else:

```
if (tipo.equals("terrestre")) transporte = new Caminhao();
else if (tipo.equals("maritimo")) transporte = new Navio();
```

Se um novo tipo de transporte for adicionado, como um drone, será preciso alterar o código existente.

Isso viola o princípio Aberto/Fechado (OCP):

“O código deve estar aberto para extensão, mas fechado para modificação.”

# SOLUÇÃO FACTORY METHOD

- ✓ Encapsular a lógica de criação em uma classe fábrica.
- ✓ Delegar às subclasses a decisão de qual objeto criar.
- ✓ O código cliente nunca precisa saber qual classe concreta está sendo usada.

# SOLUÇÃO FACTORY METHOD

```
public abstract class Logistica {  
    public abstract Transporte criarTransporte();  
  
    public void planejarEntrega() {  
        Transporte t = criarTransporte();  
        t.entregar();  
    }  
}
```

- Cria uma classe base, chamada por exemplo de **Logistica**, que define um método abstrato chamado **criarTransporte()**.
- As subclasses concretas — como **LogisticaTerrestre** e **LogisticaMaritima** — vão sobrescrever esse método e decidir qual tipo de transporte criar.
- O código cliente só chama **criarTransporte()**, sem precisar saber se o transporte será um Caminhão, um Navio ou outro tipo.

# ESTRUTURA

De forma geral, o padrão Factory Method possui quatro elementos principais:

- **Produto (Product)**: é a interface comum dos objetos criados.
- **Produto Concreto (Concrete Product)**: são as classes que implementam essa interface.
- **Criador (Creator)**: é a classe que declara o Factory Method.
- **Criadores Concretos (Concrete Creators)**: são as subclasses que implementam o método fábrica.

Essa estrutura deixa o sistema mais flexível, organizado e fácil de manter.

# VANTAGENS

- Encapsulamento
- Extensibilidade
- Baixo Acoplamento
- Flexibilidade

# DESVANTAGENS

- Complexidade Adicional
- Dificuldade de Depuração
- Pode aumentar o número de classes.

# CONCLUSÃO

- O Factory Method é uma solução poderosa para criar objetos de forma flexível e desacoplada.
- Permite que o código crie objetos de maneira inteligente, delegando a lógica de instanciação para as subclasses, sem depender diretamente das classes concretas.