

## Atividade 05 | Padrão de Projeto: Factory Method

Alunos:

- Rafael Manso Campigotto - RA: 22014205-2
- João Henriue Salvalagio - RA: 23123581-2
- Eduardo Navarro - RA: 22014553-2
- Eduardo Voltatone - RA: 22207439-2
- Luciano Giraldi - RA: 22228890-2
- Pedro Gomes - RA: 22087525-2

Projeto: FazendaPro

### Descrição da funcionalidade onde será aplicado o padrão

O projeto FazendaPro é um sistema de gerenciamento agropecuário voltado para a pecuária de corte. Uma das funcionalidades do sistema é o registro de movimentações do gado, como nascimentos, vendas e mortes.

Para tornar essa parte do código mais flexível, modular e aderente aos princípios de boas práticas da orientação a objetos, foi decidido aplicar o padrão de projeto Factory Method para criar dinamicamente os tipos de movimentações com base em uma entrada do usuário.

Esse padrão facilita a expansão para novos tipos de movimentação futuramente (ex: transferência de fazenda), sem alterar o código existente, apenas criando novas classes concretas.

### Criação dos componentes do padrão Factory

#### 1. Interface (Produto Abstrato) - Movimentacao

```
export interface Movimentacao {  
  executar(): string;  
}
```

#### 2. Produtos Concretos - Nascimento

```
export class Nascimento implements Movimentacao {  
  executar(): string {  
    return 'Nascimento registrado com sucesso.';  
  }  
}
```

## Atividade 05 | Padrão de Projeto: Factory Method

```
}  
}
```

### 2. Produtos Concretos - Venda

```
export class Venda implements Movimentacao {  
  executar(): string {  
    return 'Venda registrada com sucesso.';  
  }  
}
```

### 2. Produtos Concretos - Morte

```
export class Morte implements Movimentacao {  
  executar(): string {  
    return 'Morte registrada com sucesso.';  
  }  
}
```

### 3. Creator (Factory) - MovimentacaoFactory

```
export class MovimentacaoFactory {  
  static criar(tipo: string): Movimentacao {  
    switch (tipo.toLowerCase()) {  
      case 'nascimento':  
        return new Nascimento();  
      case 'venda':  
        return new Venda();  
      case 'morte':  
        return new Morte();  
      default:  
        throw new Error('Tipo de movimentacao invalido.');  
    }  
  }  
}
```