

## **UNIVERSIDADE POSITIVO**

### **AVALIAÇÃO SEMESTRAL**

**Prof Escobar**

**Nota máxima: 5,0**

#### **Instruções para realização:**

1. Esta avaliação possui duas partes. Uma relativa a implementação de código e outra relativa a teoria sobre padrões de projetos.
2. Para a parte relativa à implementação, entregue o código criado no Black Board
3. Para a parte teórica, responda neste mesmo documento e entregue junto com o seu código

#### **PARTE 1: Implementação**

##### **1. (2,5 pontos) Sistema de Seleção de Transporte para Entregas**

Uma empresa está desenvolvendo um sistema de e-commerce que precisa gerenciar diferentes tipos de transporte para realizar entregas aos clientes. Você foi encarregado de criar um componente para gerenciar a escolha do tipo de transporte. Este componente deve atender aos seguintes requisitos:

1. O sistema deve suportar múltiplos tipos de transporte, incluindo transporte terrestre, aéreo, marítimo e entrega por drones.
2. Cada tipo de transporte tem regras específicas de operação. Por exemplo:
  - Transporte terrestre requer a seleção de uma transportadora local.
  - Transporte aéreo deve considerar limites de peso e dimensões das encomendas.
  - Transporte marítimo é utilizado apenas para entregas internacionais.
  - Entrega por drones é limitada a áreas metropolitanas e pequenos pacotes.
3. Novos tipos de transporte podem ser adicionados futuramente sem que o código existente seja modificado.
4. O sistema deve permitir selecionar dinamicamente o tipo de transporte com base nas características da encomenda e na localização do cliente.

Implemente uma solução para atender a esses requisitos, garantindo que o sistema seja flexível, modular e mantenha boas práticas de design.

## PARTE 2: Teoria sobre padrões de projetos

2. (1,5 pontos) Associe cada conceito da Coluna A com sua respectiva descrição na Coluna B.

### Coluna A

1. Singleton
2. Injeção de Dependência
3. Interfaces
4. Decorator
5. Factory Method
6. Strategy
7. Prototype

### Coluna B

- (3) Define um contrato comum que várias classes podem implementar, garantindo um comportamento uniforme.
- (1) Garante que uma classe tenha apenas uma única instância acessível globalmente durante toda a execução do programa.
- (4) Permite adicionar funcionalidades a um objeto dinamicamente, sem modificar sua estrutura original.
- (5) Facilita a criação de objetos, delegando a responsabilidade da instância para subclasses.
- (6) Permite a seleção de diferentes comportamentos ou algoritmos em tempo de execução, encapsulando-os em classes separadas.
- (2) Reduz o acoplamento entre classes, fornecendo dependências de forma externa em vez de criá-las dentro da classe.
- (7) Cria novos objetos copiando uma instância existente, em vez de instanciá-los diretamente.

**3. (1,0 ponto) Leia as afirmações abaixo sobre os padrões de projeto Singleton, Strategy e Factory Method e indique se cada uma é Verdadeira (V) ou Falsa (F):**

(V) O padrão *Singleton* é utilizado para garantir que uma classe tenha apenas uma única instância durante a execução do programa e fornece um ponto de acesso global a essa instância.

(V) O padrão *Strategy* permite que algoritmos diferentes sejam implementados de forma intercambiável em tempo de execução, encapsulando-os em classes separadas.

(F) O padrão *Factory Method* elimina completamente a necessidade de subclasses na criação de objetos específicos.

(F) O padrão *Singleton* é adequado para situações em que é necessário garantir a criação de múltiplas instâncias independentes de uma classe.

(V) O padrão *Strategy* promove o princípio de abertura/fechamento, permitindo a extensão do comportamento sem modificar classes existentes.

(V) O padrão *Factory Method* delega a criação de objetos para subclasses, permitindo a especialização da lógica de criação.

(V) O padrão *Singleton* é adequado para gerenciar estados globais compartilhados em uma aplicação, mas deve ser usado com cuidado em aplicações multithread.