**Programação Orientada a Objetos no projeto de Loja Online**

A Programação Orientada a Objetos é baseada em quatro pilares principais: abstração, encapsulamento, herança e polimorfismo. Esses conceitos são fundamentais para o desenvolvimento de software modular e reutilizável.

**Abstração**

Abstração é a simplificação de objetos complexos, focando apenas nos detalhes essenciais. No projeto, criei classes como `Cliente`, `Produto` e `Pedido`, cada uma representando um aspecto da loja. Por exemplo, a classe `Produto` abstrai os atributos e métodos principais de um produto, como nome, código e preço.

**Encapsulamento**

Encapsulamento é proteger os dados internos de uma classe, permitindo acesso apenas por meio de métodos definidos. No projeto, usei modificadores de acesso (`private`, `public`) para controlar o acesso aos atributos, como no caso dos produtos e clientes. A classe `Loja`, por exemplo, tem métodos para adicionar e listar produtos, mas os detalhes internos estão encapsulados.

**Herança**

Herança permite a criação de novas classes com base em outras. No projeto, criei uma classe base `Produto` e, a partir dela, `ProdutoFisico` e `ProdutoDigital`. Ambas as subclasses herdaram os atributos e métodos comuns da classe `Produto`, mas cada uma possui suas particularidades.

**Polimorfismo**

O polimorfismo permite tratar objetos de classes derivadas de forma uniforme. No projeto, utilizei isso ao adicionar tanto `ProdutoFisico` quanto `ProdutoDigital` a uma lista de `Produto` no `Pedido`. Assim, o sistema consegue manipular diferentes tipos de produto sem precisar de código específico para cada um.

Com esses pilares, o projeto de loja online ficou mais organizado, reutilizável e fácil de manter. A abstração simplificou a modelagem das entidades, o encapsulamento protegeu os dados, a herança evitou repetição de código, e o polimorfismo trouxe flexibilidade ao tratar diferentes tipos de produtos.