

Padrão Facade

Padrão Estrutural

I. Problema: Fachada para Realização de um Pedido online

Um sistema de comércio eletrônico em desenvolvimento precisa lidar com o fluxo de realização de um pedido online para os clientes. Nessa interação, estão previstas as seguintes atividades:

- Validação dos dados do pedido recebidos pela interface do usuário.
- Verificação da disponibilidade dos produtos no estoque.
- Cálculo do valor total do pedido.
- Persistência do pedido no banco de dados.
- Envio de um e-mail para o cliente.

Um desenvolvedor, então, teve a ideia de encapsular todas as tarefas do fluxo de um pedido em uma única classe fachada, buscando simplificar o uso do sistema. O código a seguir ilustra a solução inicialmente proposta:

```
public class PedidoFacade {

    private EstoqueRepositorio estoqueRepositorio = new EstoqueRepositorio();
    private PedidoRepositorio pedidoRepositorio = new PedidoRepositorio();
    private EmailServico emailServico = new EmailServico();

    public void processarPedido(String nomeCliente, String email,
                                List<Item> itens) {

        // Validação básica
        if (nomeCliente == null || email == null || itens == null ||
            itens.isEmpty()) {
            throw new IllegalArgumentException("Dados do pedido inválidos");
        }

        // Verifica estoque
        for (Item item : itens) {
            if (!estoqueRepositorio.temEstoque(item.getProdutoId(),
                                                item.getQuantidade())) {
                throw new RuntimeException("Produto sem estoque: " +
                                           item.getProdutoId());
            }
        }
    }
}
```

```
// Calcula total
double total = 0;
for (Item item : itens) {
    total += item.getPrecoUnitario() * item.getQuantidade();
}

// Cria e salva pedido
Pedido pedido = new Pedido(nomeCliente, email, itens, total);
pedidoRepositorio.salvar(pedido);

// Envia e-mail
emailServico.enviarConfirmacao(pedido);
}
}
```

II. Tarefa 1

Com base no código apresentado, faça o que se pede:

- a) Aponte os problemas/violações identificados no código.

III. Tarefa 2

Para mitigar os problemas da solução inicial, a equipe resolveu trocar o método que faz tudo por métodos individuais na própria fachada.

```
public class PedidoFacade {

    public void processarPedido( String nomeCliente, String email,
                                List<Item> itens) {
        validarDados(nomeCliente, email, itens);
        verificarEstoque(itens);
        double total = calcularTotal(itens);
        Pedido pedido = criarPedido(nomeCliente, email, itens, total);
        salvarPedido(pedido);
        enviarEmail(pedido);
    }

    private void validarDados(String nomeCliente, String email,
                              List<Item> itens){
        /* ... */
    }
    private void verificarEstoque(List<Item> itens) { /* ... */ }
    private double calcularTotal(List<Item> itens) { /* ... */ }
    private Pedido criarPedido(...) { /* ... */ }
    private void salvarPedido(Pedido pedido) { /* ... */ }
    private void enviarEmail(Pedido pedido) { /* ... */ }
}
```

A ideia é dividir o método **processarPedido()** em vários métodos auxiliares dentro da mesma classe fachada, melhorando a legibilidade. Sendo assim, responda:

- O que melhora ao investir nessa abordagem?
- O que ainda precisa ser melhorado? Comente.
- Apresente um design final para o problema discutido.