

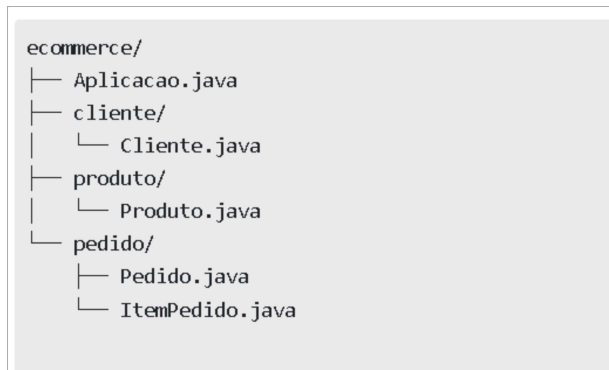
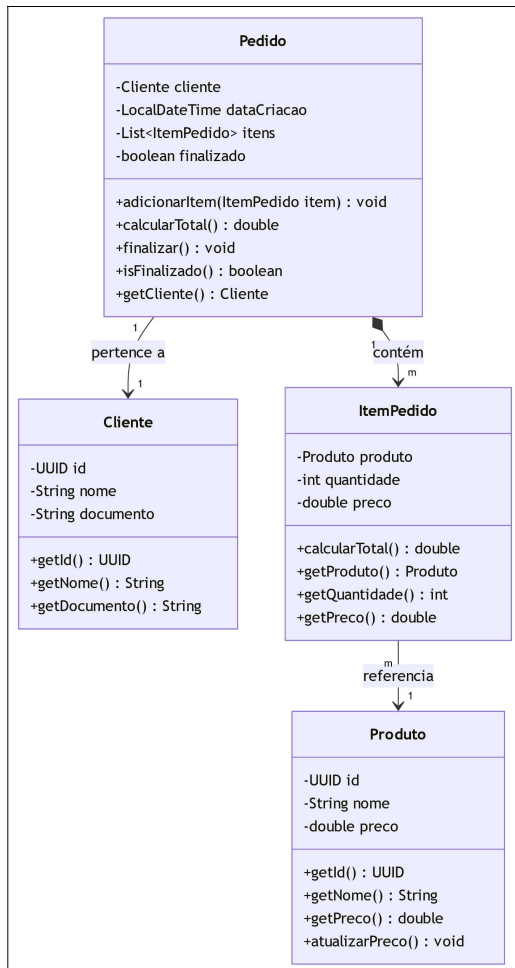
TRABALHO N°2 “ E-COMMERCE “

Nelson Alejandro Herrera Gómez

Introdução

O desenvolvimento seguiu uma progressão em três etapas que permitiram compreender a complexidade do problema: iniciando com POO básica, avançando para a implementação de interfaces e finalmente atendendo a todos os requisitos especificados para este trabalho.

a) Proposta Básica de POO + SOLID



Saida esperada :

Cliente: Pedro

Total do pedido: R\$ 6640.0

Pedido finalizado? true

b) Proposta Interfaces + Generics IRepository <T, ID>

SOLID aplicado:

- **SRP** → cada classe tem uma responsabilidade.
- **OCP** → possível criar outro repositório
- **LSP** → qualquer implementação de IRepository pode substituir outra.
- **ISP** → IServicoNotificacao
- **DIP** → ServicoPedido depende de interfaces



```
package aplicacao;
import dominio.*;
import repositorio.*;
import servico.*;
import java.util.UUID;
```

```
public class Main {
    public static void main(String[] args) {
        IRepositoryCliente repoCliente = new RepositorioClienteEmMemoria();
        IRepositoryProduto repoProduto = new RepositorioProdutoEmMemoria();
        IRepositoryPedido repoPedido = new RepositorioPedidoEmMemoria();
        IServicoNotificacao notificacao = new ServicoNotificacaoEmail();
        ValidacaoPedido validacao = new ValidacaoPedido();

        ServicoPedido servicoPedido = new ServicoPedido(repoCliente, repoProduto, repoPedido, notificacao,
        validacao);

        Cliente cliente = new Cliente("Nelson Herrera");
        Produto produto = new Produto("Notebook", 3500);

        repoCliente.salvar(cliente);
        repoProduto.salvar(produto);

        Pedido pedido = servicoPedido.criarPedido(cliente.getId());
        servicoPedido.adicionarItem(pedido.getId(), produto.getId(), 1);
        servicoPedido.finalizarPedido(pedido.getId());
    }
}
```

Salida Esperada :

Enviando e-mail para Nelson Herrera: Seu pedido foi finalizado! Valor: R\$ 3500.0

c) Proposta Interfaces + Generics + Solid

Caso de Uso - " Realizar Compra no E-Commerce "

Ator principal: Cliente

Objetivo: Realizar a compra de um produto até a entrega final

Pré-condições:

- Cliente deve estar cadastrado com **documento de identificação válido**.
- Produto deve estar cadastrado na base.

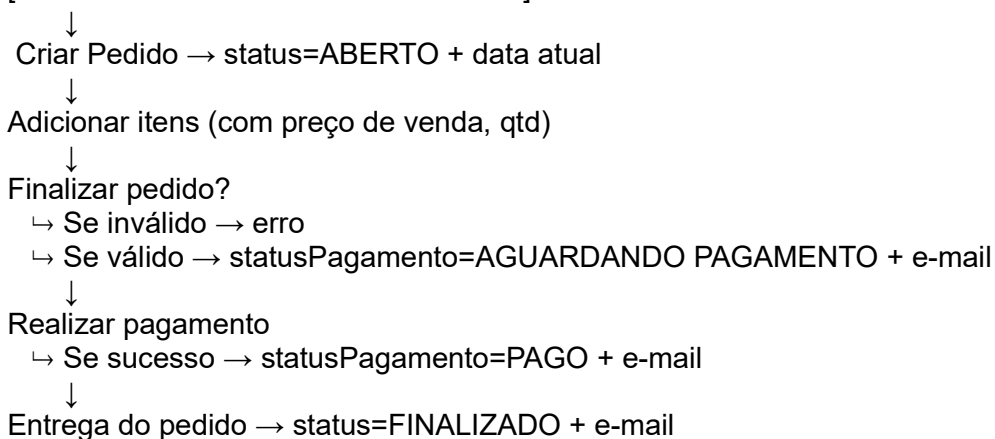
Fluxo Principal:

1. Cliente inicia um pedido.
 - Sistema gera automaticamente **data de criação**.
 - Pedido é criado com **status = ABERTO**.
2. Cliente adiciona itens ao pedido:
 - É informado **produto, quantidade e valor de venda** (pode ser diferente do valor padrão do produto).
 - Enquanto o pedido estiver com status "ABERTO", o cliente pode:
 - **Adicionar** itens.
 - **Alterar quantidade**.
 - **Remover** itens.
3. Cliente solicita **finalizar pedido**:
 - O sistema valida se existe **ao menos 1 item** e se o **valor total > 0**.
 - Caso válido: status do **pagamento = AGUARDANDO PAGAMENTO**.
 - Cliente é **notificado por e-mail**.
4. Cliente realiza o **pagamento**:
 - Somente permitido se o pedido estiver com status "AGUARDANDO PAGAMENTO".
 - Após sucesso: status do **pagamento = PAGO**.
 - Cliente é notificado.
5. Sistema realiza a **entrega**:
 - Status do pedido é alterado para **FINALIZADO**.
 - Cliente recebe notificação de entrega.

Fluxo Alternativo:

- Se o cliente tentar finalizar sem itens → erro: "Pedido inválido".
- Se tentar pagar sem estar em "Aguardando Pagamento" → erro: "Pagamento não permitido".

[Cliente cadastrado + documento válido]



Saída Esperada :

✅ Pedido criado com ID: e2e154ce-58b2-4717-a0fe-f38fe626fd96 | Status: ABERTO | Data: 2025-09-07T21:30:32.347523800

🛒 Produto adicionado: Notebook | Quantidade: 1 | Total pedido: R\$ 3500.0

✉️ E-mail para Carlos: Pedido finalizado! Valor: R\$ 3500.0

📌 Status do pagamento: AGUARDANDO PAGAMENTO

✉️ E-mail para Carlos: Pagamento confirmado do pedido e2e154ce-58b2-4717-a0fe-f38fe626fd96

💳 Pagamento realizado. Status pagamento: PAGO

✉️ E-mail para Carlos: Pedido entregue! Obrigado pela compra.

📄 Pedido entregue. Status final: FINALIZADO