

Teste Técnico – Desenvolvedor C#

Objetivo

Desenvolver uma aplicação desktop em **C# com WPF (.NET Framework 4.6)** para **cadastro e manipulação de dados** relacionados a pessoas, produtos e pedidos. O foco é avaliar a capacidade de estruturar uma aplicação com múltiplas telas, persistência de dados em arquivos, uso de LINQ e boas práticas de desenvolvimento.

Entidades e Regras de Negócio

1. Pessoa

- **Id:** preenchimento automático (somente leitura)
- **Nome:** obrigatório | campo de pesquisa
- **CPF:** obrigatório | validar CPF | campo de pesquisa
- **Endereço:** opcional

2. Produto

- **Id:** preenchimento automático (somente leitura)
- **Nome:** obrigatório | campo de pesquisa
- **Código:** obrigatório | campo de pesquisa
- **Valor:** obrigatório | pesquisa por faixa (valor inicial e final)

3. Pedido

- **Id:** preenchimento automático (somente leitura)
 - **Pessoa:** obrigatório (relacionamento)
 - **Produtos:** obrigatório (lista de produtos com quantidade)
 - **Valor Total:** calculado automaticamente
 - **Data da Venda:** preenchido automaticamente com a data atual
 - **Forma de Pagamento:** obrigatório (dinheiro, cartão, boleto)
 - **Status:** preenchido automaticamente como "Pendente"; pode ser alterado para "Pago", "Enviado" ou "Recebido"
-

Requisitos Técnicos

1. Persistência de Dados

- Os dados devem ser armazenados em **arquivos XML ou JSON**
- A manipulação dos dados deve ser feita utilizando **LINQ**

2. Interface Gráfica

- A aplicação deve conter **três telas principais**:
 - Cadastro de Pessoas
 - Cadastro de Produtos
 - Cadastro de Pedidos

3. Funcionalidades por Tela

a. Tela de Pessoas

- Filtros: Nome, CPF
- Grid com todos os registros
- Ações: Incluir, Editar, Salvar, Excluir
- Botão: **Incluir Pedido**
 - Abre a tela de pedidos vinculada à pessoa
 - Permite adicionar múltiplos produtos com quantidade
 - Botão **Finalizar Pedido**:
 - Calcula o valor total
 - Define status como "Pendente"
 - Salva o pedido
 - Se sair da tela sem finalizar, o pedido é descartado
 - Após finalizado, o pedido não pode ser alterado
- Grid de Pedidos da Pessoa:
 - Campos: Data da Venda, Valor Total, Forma de Pagamento, Status
 - Ações por linha:
 - Marcar como Pago
 - Marcar como Enviado
 - Marcar como Recebido
 - Filtros adicionais:
 - Mostrar apenas pedidos entregues
 - Mostrar apenas pedidos pagos
 - Mostrar apenas pedidos pendentes de pagamento

b. Tela de Produtos

- Filtros: Nome, Código, Faixa de Valor
- Grid com todos os registros
- Ações: Incluir, Editar, Salvar, Excluir

c. Tela de Pedidos

- Seleção de Pessoa
- Adição de múltiplos produtos com quantidade

- Cálculo automático do valor total
- Seleção da forma de pagamento
- Botão Finalizar (salva e bloqueia edição)

Critérios de Avaliação

CRITÉRIO	PESO
ORGANIZAÇÃO DO CÓDIGO E USO DE BOAS PRÁTICAS	20%
USO CORRETO DE LINQ PARA MANIPULAÇÃO DE DADOS	20%
PERSISTÊNCIA CORRETA EM XML OU JSON	20%
FUNCIONALIDADE COMPLETA DAS TELAS E REGRAS DE NEGÓCIO	30%
INTERFACE AMIGÁVEL E USABILIDADE	10%

Entrega

- Enviar o projeto completo em um repositório Git (GitHub, GitLab, etc.)
- Incluir um **README** com instruções de execução e qualquer dependência necessária

Estrutura de Pastas Sugerida

WpfApp/

— Models/	# Classes de domínio (Pessoa, Produto, Pedido)
— Views/	# Telas WPF (XAML)
— ViewModels/	# Lógica de apresentação (MVVM)
— Services/	# Serviços de persistência e lógica de negócio
— Data/	# Arquivos JSON/XML
— Resources/	# Ícones, imagens, etc.
— README.md	# Instruções do projeto