

## Teste Técnico – Gerenciamento de Pedidos

Desenvolva uma aplicação em C# para gerenciar pedidos de clientes. A solução deve atender aos seguintes requisitos:

---

### Requisitos Funcionais:

1. Crie um CRUD (Create, Read, Update, Delete) para gerenciar pedidos.
  2. Cada pedido deve conter:
    - Identificador único (ID).
    - Nome do cliente.
    - Data do pedido.
    - Itens do pedido (nome do item, quantidade, valor unitário).
    - Valor total do pedido (calculado com base nos itens).
  3. Permitir filtrar os pedidos por data e/ou nome do cliente.
- 

### Requisitos Técnicos:

1. Utilize **.NET 6** ou superior.
  2. Armazene os dados em memória (usando uma lista) ou em um banco de dados local (SQLite, por exemplo).
  3. Exponha uma API REST com os seguintes endpoints:
    - POST /orders: Criar um pedido.
    - GET /orders: Listar todos os pedidos.
    - GET /orders/{id}: Obter um pedido por ID.
    - PUT /orders/{id}: Atualizar um pedido.
    - DELETE /orders/{id}: Excluir um pedido.
  4. Implemente testes unitários usando **xUnit**.
- 

### Extras (não obrigatório, mas diferencial):

1. Documentação da API usando **Swagger**.
2. Validação dos dados de entrada (por exemplo, não permitir valores negativos para quantidade ou preço).
3. Deploy no **Azure App Service** ou outra plataforma de sua escolha.
4. **Use MediatR:**

5. Utilizar Contêiner
  6. Clean Architecture
  7.
    - Implemente padrões **CQRS** para separar comandos e queries.
    - Use **Requests** e **Handlers** do MediatR para processar as operações do CRUD.
  8. Configure **injeção de dependência** e modularize o projeto para facilitar a escalabilidade.
- 

#### Instruções para Entrega:

1. Crie um repositório no **GitHub**.
  2. Estructure o projeto de forma clara, incluindo pastas como src, tests, e docs (se necessário).
  3. Inclua um arquivo README.md com:
    - Descrição do projeto.
    - Instruções para rodar a aplicação localmente.
    - Exemplos de chamadas aos endpoints.
  4. Suba o código no GitHub e compartilhe o link do repositório.
  5. Prazo 4 dias
- 

**Envie o endereço do repositório para a consultoria**