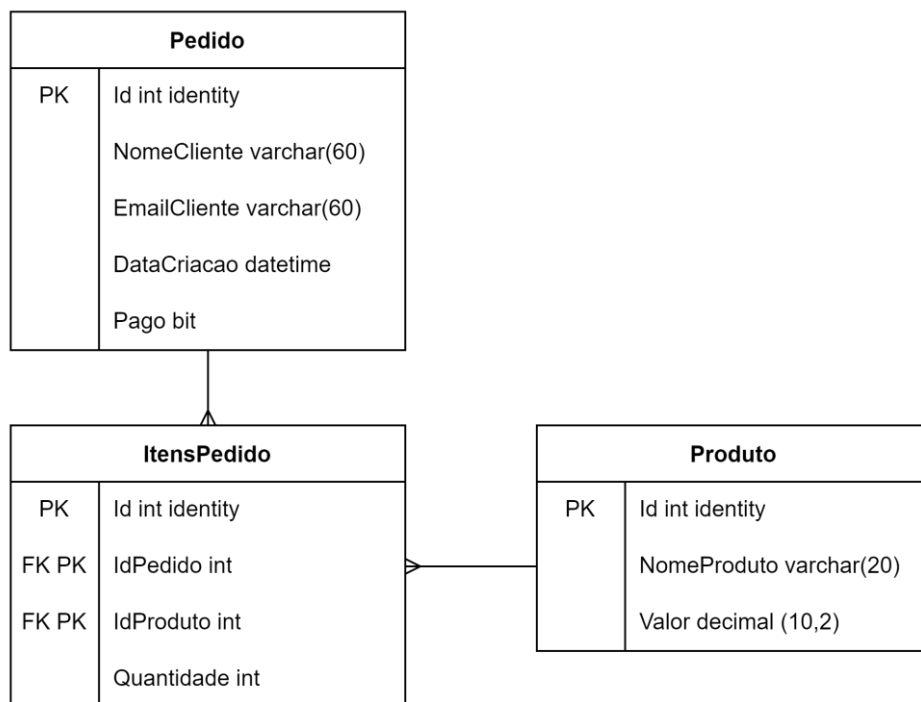


Desafio para candidatos da Stefanini (v3)

O desafio consiste em criar uma **API** em **.NET**, onde você deve desenvolver o **CRUD** de pedidos conforme a modelagem de dados fornecida. A **base de dados** utilizada deve ser o **SQL Server**.

É necessário implementar, no mínimo, testes unitários para o método **GET** de pedidos, abrangendo tanto a **camada de visualização** quanto a **camada de serviço**.

Abaixo segue a modelagem de dados que deve ser implementado no CRUD.



Arquitetura

A API deverá ser desenvolvida em **.NET**, utilizando a arquitetura de sua escolha, sendo a **Clean Architecture** preferida. É fundamental aplicar princípios como **Domain-Driven Design (DDD)**, **SOLID**, **Clean Code** e garantir a conformidade com as melhores práticas de **APIs REST**. Espera-se um código bem estruturado, de fácil manutenção e extensível, com foco em modularidade e reutilização.

O front-end pode ser desenvolvido utilizando qualquer framework ou biblioteca JavaScript de qualquer versão, porém temos preferência por **Angular**.

Entregáveis

A não entrega do obrigatório o candidato será desclassificado.

Obrigatório

- Projeto compilando com sucesso;
- Conceitos de DDD, Solid e Clean Code;
- API .NET com CRUD de criar **PEDIDO**;
- Teste unitário back-end no **mínimo o GET Pedido**;
- Migration SQL funcionando no projeto;
- Swagger implementado e funcionando;
- O GET de PEDIDO deve retornar o modelo de JSON (JSON 1);
- Projeto publicado em repositório GIT.

Desejável

- Arquivo README com informações sobre a aplicação;
- Front-end com todas as integrações e os comportamentos básicos funcionando;
- API e Front-end publicada em Cloud;
- Banco de dados no docker compose
- Design Patterns

```
JSON GET PEDIDOS
{
  "id": 0,
  "nomeCliente": "",
```

```
"emailCliente": "",
"pago": true,
"valorTotal": 0.0, (Soma dos produtos incluso no pedido, considerando a quantidade comprada)
"itensPedido": [
  {
    "id": 0,
    "idProduto": 0,
    "nomeProduto": "",
    "valorUnitario": 0.0,
    "quantidade": 0
  }
]
}
```

JSON 1 – Modelo GET Pedidos

Requisitos do ambiente

- Visual Studio 2022 ou Visual Studio Code;
- Banco de dados SQL, InMemory (detalhes abaixo) ou via Container (SQL Server).
 - Pode-se utilizar a lib do Entity Framework Core de banco de dados em memória oficial Microsoft.

Mais Informações:

<https://learn.microsoft.com/pt-br/ef/core/providers/in-memory/?tabs=dotnet-core-cli>