Perfeito! Aqui está uma versão refinada e profissional da descrição do projeto, incluindo as instruções de execução e a imagem fornecida:

Ambev Developer Evaluation - API de Vendas



🙎 Desenvolvido por

Edio Rhoden

🕒 Projeto entregue em 1 dia e meio como parte de uma avaliação técnica para desenvolvedores .NET.

Visão Geral

Esta aplicação representa uma API RESTful moderna, desenvolvida em .NET 8, para o gerenciamento completo de um sistema de vendas, com foco em boas práticas de arquitetura e código limpo.

Ela implementa as operações fundamentais de um ciclo de vendas, com suporte para criação, atualização, exclusão e consulta de:

- Vendas
- **Produtos**
- Clientes
- **Filiais**

Tecnologias e Padrões Utilizados

Tecnologia/Padrão	Objetivo
.NET 8	Plataforma principal da API
Entity Framework Core	Persistência de dados com banco InMemory
MediatR	Implementação do padrão CQRS
AutoMapper	Mapeamento de entidades e DTOs
FluentValidation	Validação desacoplada e robusta
Swagger (Swashbuckle)	Documentação interativa da API

Tecnologia/Padrão Objetivo

Serilog Log estruturado e eventos customizados

Clean Architecture + DDD Separação clara de responsabilidades

Docker (parcialmente) Containerização pronta para extensão futura

Módulos da API

A solução está organizada em camadas e pastas conforme o padrão **DDD (Domain- Driven Design)**:

src/

— Ambev.DeveloperEvaluation.WebApi // API pública e controllers

— Ambev.DeveloperEvaluation.Application // CQRS: Commands, Queries, Handlers

— Ambev.DeveloperEvaluation.Domain // Entidades e interfaces

— Ambev.DeveloperEvaluation.ORM // Contexto EF, Repositórios, Seeders

├— Ambev.DeveloperEvaluation.IoC // Injeção de dependências

├— Ambev.DeveloperEvaluation.Infrastructure// Publisher de eventos por log

Funcionalidades por Módulo

Vendas (/api/sale)

- Criação de vendas com itens e regras de desconto
- Cálculo automático de total
- Publicação de eventos de domínio (SaleCreated, SaleModified)
- Suporte a cancelamento e exibição do histórico

Produtos (/api/products)

- Cadastro, consulta, atualização e remoção
- Nome, descrição e preço unitário

Clientes (/api/customers)

• Cadastro, consulta, atualização e remoção

• Nome, e-mail, telefone

Filiais (/api/branches)

- Cadastro, consulta, atualização e remoção
- Nome e localização

Regras de Negócio

Quantidade Desconto Aplicado

1 a 3 Nenhum

4 a 9 10%

10 a 20 20%

+20 Não permitido

Como Executar o Projeto

Etapas para rodar localmente:

- 1. Clone o repositório:
- 2. git clone https://github.com/ediorhodendev/Ambev.git
- 3. Abra o projeto no Visual Studio ou VS Code.
- 4. Execute a solução escolhendo o projeto Ambev.DeveloperEvaluation.WebApi como *startup project*.
- 5. Utilize a opção **IIS Express** conforme a imagem abaixo para executar facilmente:



Banco de Dados

O sistema utiliza **EF Core InMemory**, e **ao executar a aplicação**, **os seguintes dados são automaticamente criados**:

- 30 produtos fictícios
- 30 clientes
- 30 filiais
- Vendas geradas automaticamente com itens e descontos aplicados

Esses dados são úteis para testes e demonstrações via Swagger ou Postman.

■ Interface Swagger

Acesse https://localhost:<porta>/swagger para interagir com a API por meio de uma interface intuitiva, que exibe todos os endpoints e modelos de dados.

Eventos Publicados

A API publica eventos simulados via logs para futuras integrações com mensageria:

- SaleCreated
- SaleModified
- SaleCancelled
- ItemCancelled (planejado)

Exemplo:

[EVENTO] SaleCreated: { Id = ..., Total = ..., Customer = ..., Items = [...] }

> Considerações Finais

- Projeto finalizado em tempo recorde: 1 dia e meio
- Arquitetura limpa e escalável
- Pronto para ser conectado a banco relacional (PostgreSQL, SQL Server)
- Pode ser evoluído para mensageria real, autenticação JWT, fila de pedidos e muito mais.