Planejamento de Integrações – eSellers (.NET)

# Análise atual do projeto

* - Frontend estruturado com autenticação e gestão de produtos.
* - Backend em .NET ainda sem integrações com marketplaces ou SEFAZ.

# Alterações recomendadas agora

* 1. Criar base de diretórios para integrações no backend .NET.
* 2. Criar modelos internos padronizados como MarketplaceProduct e Invoice.
* 3. Evitar implementar lógica de integração no frontend – use API do backend.

# Próximas etapas (por sprints)

* Sprint 1 – Integração com Marketplaces: iniciar com Mercado Livre usando OAuth 2.0.
* Sprint 2 – Integração com SEFAZ: gerar XML, assinar digitalmente e transmitir.
* Sprint 3 – Sincronização e Webhooks: jobs agendados e escuta de eventos.

# Boas práticas

* - Padronizar modelos internos de dados.
* - Criptografar tokens de APIs de terceiros.
* - Separar integrações por plataforma.
* - Registrar falhas e métricas de integração.

# Sobre a estrutura de diretórios (.NET)

* /eSellers.Application
* /Integrations
* /Marketplaces
* /MercadoLivre
* - MercadoLivreService.cs
* - MercadoLivreMapper.cs
* - MercadoLivreOAuthHandler.cs
* /Fiscal
* /SEFAZ
* - NFeService.cs
* - XmlBuilder.cs
* - CertificateHandler.cs
* /eSellers.Domain
* /Entities
* - MarketplaceProduct.cs
* - Invoice.cs
* /Interfaces
* - IMarketplaceIntegration.cs
* - IInvoiceService.cs
* /eSellers.Infrastructure
* /ExternalServices
* - MercadoLivreApiClient.cs
* - SEFAZWebServiceClient.cs

# Estudos recomendados (.NET)

* - Clean Architecture com .NET: https://dev.to/khalidabuhakmeh/clean-architecture-in-asp-net-core-2021-4ahf
* - Comunicação SEFAZ (Zeus NFe): https://github.com/adeniltonbs/Zeus.Net.NFe.NFCe
* - OAuth2 no ASP.NET: https://medium.com/@iammukeshm/oauth-2-0-authorization-with-asp-net-core-6-api-b769c1ca69e7
* - Arquiteturas modernas .NET: https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures