Aqui está a estrutura completa e detalhada do projeto ShopMaster, incluindo os serviços adicionais:

/ShopMaster

├── /CatalogService

│ ├── /API

│ │ ├── /Controllers

│ │ │ ├── ProductsController.cs

│ │ │ ├── CategoriesController.cs

│ │ │ └── ReviewsController.cs

│ │ ├── /DTOs

│ │ │ ├── ProductDto.cs

│ │ │ ├── CategoryDto.cs

│ │ │ └── ReviewDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ ├── ProductDto.cs

│ │ │ ├── CategoryDto.cs

│ │ │ └── ReviewDto.cs

│ │ ├── /Interfaces

│ │ │ ├── IProductService.cs

│ │ │ ├── ICategoryService.cs

│ │ │ └── IReviewService.cs

│ │ ├── /Mappings

│ │ │ └── ProductProfile.cs

│ │ ├── /UseCases

│ │ │ ├── AddProductUseCase.cs

│ │ │ ├── AddCategoryUseCase.cs

│ │ │ └── AddReviewUseCase.cs

│ │ └── /Services

│ │ └── ProductService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ ├── Product.cs

│ │ │ ├── Category.cs

│ │ │ └── Review.cs

│ │ ├── /ValueObjects

│ │ │ └── Price.cs

│ │ └── /Interfaces

│ │ ├── IProductRepository.cs

│ │ ├── ICategoryRepository.cs

│ │ └── IReviewRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── CatalogDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ ├── ProductRepository.cs

│ │ │ ├── CategoryRepository.cs

│ │ │ └── ReviewRepository.cs

│ │ └── /Services

│ │ └── ExternalProductService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

├── /OrderService

│ ├── /API

│ │ ├── /Controllers

│ │ │ ├── OrdersController.cs

│ │ │ └── CartController.cs

│ │ ├── /DTOs

│ │ │ ├── OrderDto.cs

│ │ │ └── CartDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ ├── OrderDto.cs

│ │ │ └── CartDto.cs

│ │ ├── /Interfaces

│ │ │ ├── IOrderService.cs

│ │ │ └── ICartService.cs

│ │ ├── /Mappings

│ │ │ └── OrderProfile.cs

│ │ ├── /UseCases

│ │ │ ├── CreateOrderUseCase.cs

│ │ │ ├── AddToCartUseCase.cs

│ │ │ └── RemoveFromCartUseCase.cs

│ │ └── /Services

│ │ └── OrderService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ ├── Order.cs

│ │ │ └── Cart.cs

│ │ ├── /ValueObjects

│ │ │ └── OrderStatus.cs

│ │ └── /Interfaces

│ │ ├── IOrderRepository.cs

│ │ └── ICartRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── OrderDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ ├── OrderRepository.cs

│ │ │ └── CartRepository.cs

│ │ └── /Services

│ │ └── ExternalOrderService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

├── /PaymentService

│ ├── /API

│ │ ├── /Controllers

│ │ │ └── PaymentsController.cs

│ │ ├── /DTOs

│ │ │ └── PaymentDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ └── PaymentDto.cs

│ │ ├── /Interfaces

│ │ │ └── IPaymentService.cs

│ │ ├── /Mappings

│ │ │ └── PaymentProfile.cs

│ │ ├── /UseCases

│ │ │ └── ProcessPaymentUseCase.cs

│ │ └── /Services

│ │ └── PaymentService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ └── Payment.cs

│ │ ├── /ValueObjects

│ │ │ └── PaymentStatus.cs

│ │ └── /Interfaces

│ │ └── IPaymentRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── PaymentDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ └── PaymentRepository.cs

│ │ └── /Services

│ │ └── ExternalPaymentService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

├── /IdentityService

│ ├── /API

│ │ ├── /Controllers

│ │ │ ├── AuthController.cs

│ │ │ └── UsersController.cs

│ │ ├── /DTOs

│ │ │ ├── LoginDto.cs

│ │ │ ├── RegisterDto.cs

│ │ │ └── UserDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ ├── LoginDto.cs

│ │ │ ├── RegisterDto.cs

│ │ │ └── UserDto.cs

│ │ ├── /Interfaces

│ │ │ ├── IAuthService.cs

│ │ │ └── IUserService.cs

│ │ ├── /Mappings

│ │ │ └── UserProfile.cs

│ │ ├── /UseCases

│ │ │ ├── LoginUseCase.cs

│ │ │ ├── RegisterUseCase.cs

│ │ │ └── GetUserProfileUseCase.cs

│ │ └── /Services

│ │ └── AuthService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ ├── User.cs

│ │ │ └── Role.cs

│ │ ├── /ValueObjects

│ │ │ └── UserRole.cs

│ │ └── /Interfaces

│ │ ├── IUserRepository.cs

│ │ └── IRoleRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── IdentityDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ ├── UserRepository.cs

│ │ │ └── RoleRepository.cs

│ │ └── /Services

│ │ └── ExternalAuthService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

├── /ShippingService

│ ├── /API

│ │ ├── /Controllers

│ │ │ └── ShippingController.cs

│ │ ├── /DTOs

│ │ │ └── ShippingDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ └── ShippingDto.cs

│ │ ├── /Interfaces

│ │ │ └── IShippingService.cs

│ │ ├── /Mappings

│ │ │ └── ShippingProfile.cs

│ │ ├── /UseCases

│ │ │ └── TrackShipmentUseCase.cs

│ │ └── /Services

│ │ └── ShippingService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ └── Shipment.cs

│ │ ├── /ValueObjects

│ │ │ └── ShipmentStatus.cs

│ │ └── /Interfaces

│ │ └── IShipmentRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── ShippingDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ └── ShipmentRepository.cs

│ │ └── /Services

│ │ └── ExternalShippingService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

├── /NotificationService

│ ├── /API

│ │ ├── /Controllers

│ │ │ └── NotificationController.cs

│ │ ├── /DTOs

│ │ │ └── NotificationDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ └── NotificationDto.cs

│ │ ├── /Interfaces

│ │ │ └── INotificationService.cs

│ │ ├── /Mappings

│ │ │ └── NotificationProfile.cs

│ │ ├── /UseCases

│ │ │ └── SendNotificationUseCase.cs

│ │ └── /Services

│ │ └── NotificationService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ └── Notification.cs

│ │ ├── /ValueObjects

│ │ │ └── NotificationStatus.cs

│ │ └── /Interfaces

│ │ └── INotificationRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── NotificationDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ └── NotificationRepository.cs

│ │ └── /Services

│ │ └── ExternalNotificationService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

├── /AnalyticsService

│ ├── /API

│ │ ├── /Controllers

│ │ │ └── AnalyticsController.cs

│ │ ├── /DTOs

│ │ │ └── ReportDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ └── ReportDto.cs

│ │ ├── /Interfaces

│ │ │ └── IAnalyticsService.cs

│ │ ├── /Mappings

│ │ │ └── AnalyticsProfile.cs

│ │ ├── /UseCases

│ │ │ └── GenerateReportUseCase.cs

│ │ └── /Services

│ │ └── AnalyticsService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ └── Report.cs

│ │ ├── /ValueObjects

│ │ │ └── ReportType.cs

│ │ └── /Interfaces

│ │ └── IReportRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── AnalyticsDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ └── ReportRepository.cs

│ │ └── /Services

│ │ └── ExternalAnalyticsService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

├── /CustomerService

│ ├── /API

│ │ ├── /Controllers

│ │ │ └── CustomerController.cs

│ │ ├── /DTOs

│ │ │ └── CustomerDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ └── CustomerDto.cs

│ │ ├── /Interfaces

│ │ │ └── ICustomerService.cs

│ │ ├── /Mappings

│ │ │ └── CustomerProfile.cs

│ │ ├── /UseCases

│ │ │ └── ManageCustomerUseCase.cs

│ │ └── /Services

│ │ └── CustomerService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ └── Customer.cs

│ │ ├── /ValueObjects

│ │ │ └── CustomerType.cs

│ │ └── /Interfaces

│ │ └── ICustomerRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── CustomerDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ └── CustomerRepository.cs

│ │ └── /Services

│ │ └── ExternalCustomerService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

├── /PromotionService

│ ├── /API

│ │ ├── /Controllers

│ │ │ └── PromotionController.cs

│ │ ├── /DTOs

│ │ │ └── PromotionDto.cs

│ │ ├── /Extensions

│ │ │ ├── SwaggerExtensions.cs

│ │ │ └── ApiVersioningExtensions.cs

│ │ └── Program.cs

│ ├── /Application

│ │ ├── /DTOs

│ │ │ └── PromotionDto.cs

│ │ ├── /Interfaces

│ │ │ └── IPromotionService.cs

│ │ ├── /Mappings

│ │ │ └── PromotionProfile.cs

│ │ ├── /UseCases

│ │ │ └── ManagePromotionUseCase.cs

│ │ └── /Services

│ │ └── PromotionService.cs

│ ├── /Domain

│ │ ├── /Entities

│ │ │ └── Promotion.cs

│ │ ├── /ValueObjects

│ │ │ └── PromotionType.cs

│ │ └── /Interfaces

│ │ └── IPromotionRepository.cs

│ ├── /Infrastructure

│ │ ├── /Data

│ │ │ ├── PromotionDbContext.cs

│ │ │ └── /Migrations

│ │ ├── /Repositories

│ │ │ └── PromotionRepository.cs

│ │ └── /Services

│ │ └── ExternalPromotionService.cs

│ ├── /Crosscutting

│ │ ├── /Logging

│ │ │ ├── ILoggerManager.cs

│ │ │ └── LoggerManager.cs

│ │ ├── /Resilience

│ │ │ ├── IResiliencePolicy.cs

│ │ │ └── ResiliencePolicy.cs

│ │ ├── /Configuration

│ │ │ └── GlobalConfiguration.cs

│ │ └── /Shared

│ │ ├── /Common

│ │ └── /Models

│ └── /Docs

│ └── Architecture.md

│

└── /Common

├── /Utils

│ ├── /Helpers

│ │ └── FileHelper.cs

│ └── /Extensions

│ └── StringExtensions.cs

├── /Models

│ ├── /BaseModel.cs

│ └── /ResponseModel.cs

└── /Interfaces

├── /ILoggerManager.cs

└── /IResiliencePolicy.cs

**Descrição Geral**

1. **CatalogService**
   * Gerencia a exibição e busca de produtos, categorias e avaliações.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para produtos, categorias e avaliações.
2. **OrderService**
   * Gerencia pedidos e carrinhos de compras.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para pedidos e carrinhos.
3. **PaymentService**
   * Gerencia pagamentos e transações financeiras.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para pagamentos.
4. **IdentityService**
   * Gerencia autenticação e autorização de usuários.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para usuários e funções.
5. **ShippingService**
   * Gerencia o rastreamento e logística de envios.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para envios e rastreamento.
6. **NotificationService**
   * Gerencia o envio de notificações aos usuários.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para notificações.
7. **AnalyticsService**
   * Analisa e gera relatórios sobre o uso e desempenho do sistema.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para relatórios e análises.
8. **CustomerService**
   * Gerencia informações e interações com clientes.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para clientes.
9. **PromotionService**
   * Gerencia promoções e descontos.
   * Inclui APIs, mapeamento, serviços, domínio e infraestrutura para promoções.
10. **Common**
    * Contém utilitários e modelos compartilhados entre os serviços.

Essa estrutura modulariza o projeto, facilitando a manutenção e escalabilidade, além de permitir a adição de novos serviços conforme necessário. Se precisar de mais detalhes ou ajustes, estou à disposição!

**Passos para Criar a Estrutura no Visual Studio 2022**

1. **Crie a Solução Principal:**
   * Abra o Visual Studio 2022.
   * Selecione "Criar um novo projeto".
   * Escolha "Solução em Branco" e clique em "Avançar".
   * Nomeie a solução como ShopMaster e selecione o local onde deseja salvar.
   * Clique em "Criar".
2. **Adicione os Projetos:** Para cada serviço e projeto auxiliar (Common), você precisará adicionar projetos individuais. Aqui está como fazer isso:
   * **Adicionar um Novo Projeto:**
     + Clique com o botão direito do mouse na solução (ShopMaster) no Solution Explorer.
     + Selecione "Adicionar" > "Novo Projeto".
     + Escolha "Aplicativo Web ASP.NET Core" para os projetos de API e "Biblioteca de Classes" para os projetos de aplicação, domínio, infraestrutura e crosscutting.
     + Nomeie o projeto de acordo com a estrutura fornecida (CatalogService, OrderService, etc.).
   * **Para cada Serviço:**
     + **API:** Adicione um projeto ASP.NET Core Web API.
     + **Application, Domain, Infrastructure, Crosscutting:** Adicione projetos de Biblioteca de Classes.
   * **Para o Projeto Common:**
     + Adicione um projeto de Biblioteca de Classes e nomeie como Common.
3. **Organize os Diretórios e Arquivos:** Após criar todos os projetos, organize os diretórios e arquivos conforme sua estrutura.
   * No Solution Explorer, expanda cada projeto e crie as pastas e arquivos conforme sua estrutura.
   * Exemplo: No projeto CatalogService.API, crie as pastas Controllers, DTOs, Extensions, e adicione os arquivos correspondentes.
4. **Configuração dos Projetos:**
   * **Referências entre Projetos:**
     + Configure referências entre projetos para que possam interagir conforme necessário. Por exemplo, CatalogService.Application pode referenciar CatalogService.Domain e CatalogService.Infrastructure.
     + Clique com o botão direito em "Dependências" de um projeto e selecione "Adicionar Referência de Projeto".
   * **Configuração de Dependências:**
     + Configure pacotes NuGet e outras dependências conforme necessário para cada projeto (e.g., Entity Framework Core, AutoMapper, etc.).
5. **Configuração do Common:**
   * O projeto Common deve conter utilitários e interfaces que são compartilhados entre os diferentes serviços.
   * Certifique-se de que os outros projetos referenciam o Common onde necessário.
6. **Documentação:**
   * Adicione o arquivo Architecture.md nas pastas Docs de cada serviço, conforme especificado.

**Criar o Projeto CatalogService**

1. **Abra o Visual Studio 2022**.
2. **Crie uma nova solução** chamada ShopMaster (se ainda não existir).
3. Dentro da solução, adicione um novo projeto ASP.NET Core Web API para o CatalogService.API.

**2. Criar a Estrutura de Pastas**

Agora, você precisa criar a estrutura de pastas conforme especificado:

**No terminal do Visual Studio ou no Windows Command Prompt/PowerShell, você pode executar comandos como abaixo para criar as pastas:**

bash

Copiar código

# Navegar para a pasta do projeto CatalogService.API

cd /path/to/ShopMaster/CatalogService.API

# Criar a estrutura de pastas

mkdir Controllers DTOs Extensions

# Navegar para o projeto Application e criar pastas

cd ../CatalogService.Application

mkdir DTOs Interfaces Mappings UseCases Services

# Navegar para o projeto Domain e criar pastas

cd ../CatalogService.Domain

mkdir Entities ValueObjects Interfaces

# Navegar para o projeto Infrastructure e criar pastas

cd ../CatalogService.Infrastructure

mkdir Data\Repositories\Services

# Navegar para o projeto Crosscutting e criar pastas

cd ../CatalogService.Crosscutting

mkdir Logging Resilience Configuration Shared

mkdir Shared\Common Shared\Models

# Criar a pasta Docs

cd ../CatalogService.Docs

touch Architecture.md

**3. Criar os Arquivos**

Após criar as pastas, crie os arquivos dentro de cada pasta conforme especificado.

1. **Controllers**: Crie os arquivos ProductsController.cs, CategoriesController.cs, e ReviewsController.cs dentro de CatalogService.API/Controllers.
2. **DTOs**: Crie os arquivos ProductDto.cs, CategoryDto.cs, e ReviewDto.cs dentro de CatalogService.API/DTOs e CatalogService.Application/DTOs.
3. **Extensions**: Crie os arquivos SwaggerExtensions.cs e ApiVersioningExtensions.cs dentro de CatalogService.API/Extensions.
4. **Application**: Crie interfaces, mappings, use cases e services conforme o nome dos arquivos indicados.
5. **Domain**: Crie as entidades, objetos de valor e interfaces dentro de CatalogService.Domain/Entities, CatalogService.Domain/ValueObjects e CatalogService.Domain/Interfaces.
6. **Infrastructure**: Crie o CatalogDbContext.cs dentro de CatalogService.Infrastructure/Data e crie as classes de repositórios dentro de CatalogService.Infrastructure/Repositories.
7. **Crosscutting**: Crie as classes relacionadas a logging, resiliência, configuração e componentes compartilhados dentro de CatalogService.Crosscutting.
8. **Docs**: Edite o arquivo Architecture.md para documentar a arquitetura do CatalogService.

**4. Configurar o Git e .gitignore**

1. **.gitignore**: Certifique-se de adicionar um arquivo .gitignore no nível da solução (ShopMaster.sln). Um .gitignore padrão para projetos .NET pode ser gerado usando o comando:

bash

Copiar código

dotnet new gitignore

1. **Commit Inicial**: Agora, faça o commit inicial de todo o projeto:

bash

Copiar código

git add .

git commit -m "Initial commit - CatalogService structure"

git push origin master

frontend

/frontend

├── /src

│ ├── /app

│ │ ├── /core

│ │ │ ├── /services

│ │ │ ├── /guards

│ │ │ └── /interceptors

│ │ ├── /features

│ │ │ ├── /catalog

│ │ │ │ ├── /components

│ │ │ │ ├── /containers

│ │ │ │ ├── /services

│ │ │ │ ├── catalog.module.ts

│ │ │ │ └── catalog-routing.module.ts

│ │ │ ├── /orders

│ │ │ │ ├── /components

│ │ │ │ ├── /containers

│ │ │ │ ├── /services

│ │ │ │ ├── orders.module.ts

│ │ │ │ └── orders-routing.module.ts

│ │ │ ├── /payments

│ │ │ │ ├── /components

│ │ │ │ ├── /containers

│ │ │ │ ├── /services

│ │ │ │ ├── payments.module.ts

│ │ │ │ └── payments-routing.module.ts

│ │ │ ├── /identity

│ │ │ │ ├── /components

│ │ │ │ ├── /containers

│ │ │ │ ├── /services

│ │ │ │ ├── identity.module.ts

│ │ │ │ └── identity-routing.module.ts

│ │ │ ├── /shipping

│ │ │ │ ├── /components

│ │ │ │ ├── /containers

│ │ │ │ ├── /services

│ │ │ │ ├── shipping.module.ts

│ │ │ │ └── shipping-routing.module.ts

│ │ │ ├── /notification

│ │ │ │ ├── /components

│ │ │ │ ├── /containers

│ │ │ │ ├── /services

│ │ │ │ ├── notification.module.ts

│ │ │ │ └── notification-routing.module.ts

│ │ │ ├── /analytics

│ │ │ │ ├── /components

│ │ │ │ ├── /containers

│ │ │ │ ├── /services

│ │ │ │ ├── analytics.module.ts

│ │ │ │ └── analytics-routing.module.ts

│ │ │ ├── /customer

│ │ │ │ ├── /components

│ │ │ │ ├── /containers

│ │ │ │ ├── /services

│ │ │ │ ├── customer.module.ts

│ │ │ │ └── customer-routing.module.ts

│ │ │ └── /promotion

│ │ │ ├── /components

│ │ │ ├── /containers

│ │ │ ├── /services

│ │ │ ├── promotion.module.ts

│ │ │ └── promotion-routing.module.ts

│ ├── /assets

│ │ ├── /images

│ │ └── /styles

│ ├── /environments

│ │ ├── environment.ts

│ │ └── environment.prod.ts

│ ├── /shared

│ │ ├── /components

│ │ ├── /directives

│ │ ├── /pipes

│ │ ├── /models

│ │ └── /services

│ ├── /styles

│ │ └── styles.scss

│ ├── index.html

│ ├── main.ts

│ ├── polyfills.ts

│ ├── angular.json

│ ├── package.json

│ └── tsconfig.json

├── /e2e

│ ├── /src

│ │ └── app.e2e-spec.ts

│ ├── protractor.conf.js

│ └── tsconfig.json

└── README.md