Manual de Implementação da Arquitetura

Este documento tem como objetivo além de apresentar a arquitetura do sistema que será implementado, mostrar o passo a passo de como implementar novos recursos.

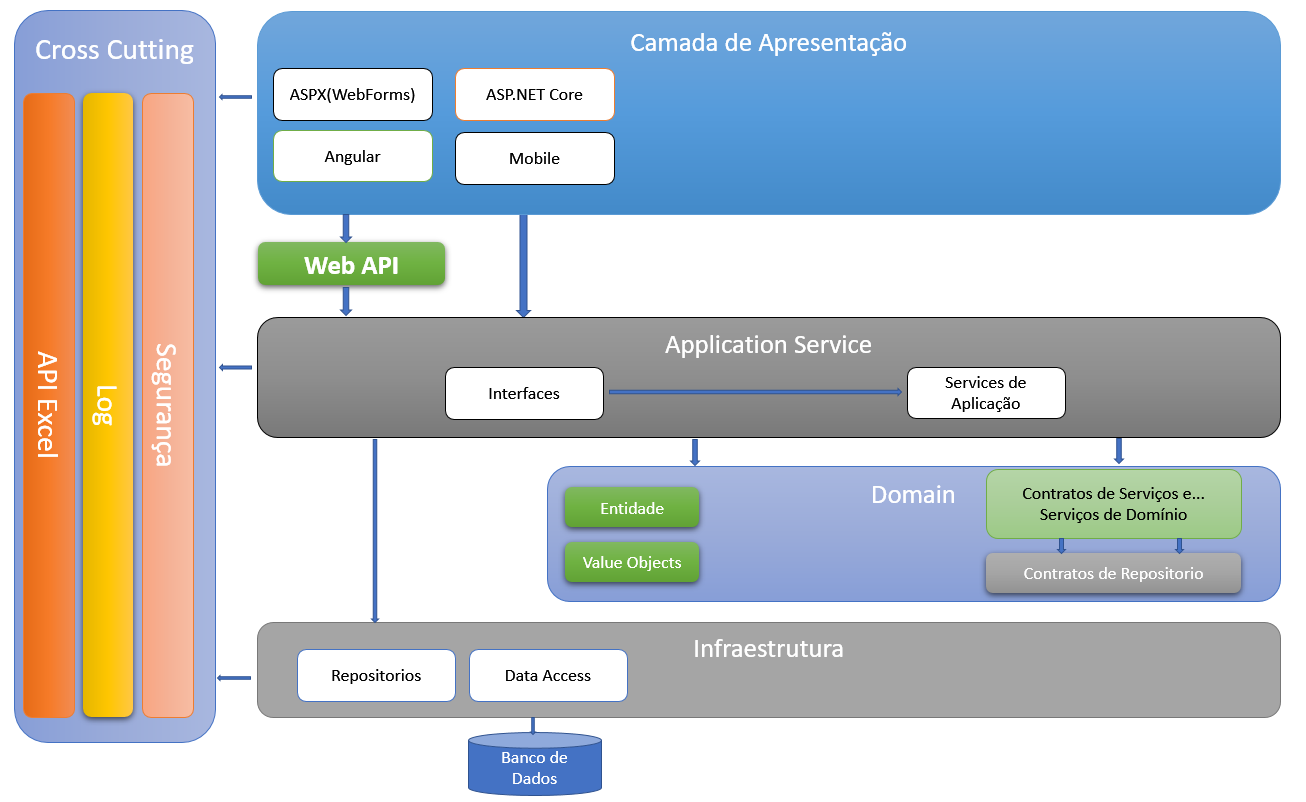
A arquitetura do sistema tem forte influência da abordagem **Domain Driven Design** e este é um assunto extenso e teoricamente deveria ser consumido antes mesmo de pensar em aprender como implementá-lo.

Mas... levando em consideração que o material original criado pelo **Eric Evans** além de denso não é nada prático, utilizando este material você pode aprender logo como implementar um sistema utilizando o DDD e depois você pode, caso queira, buscar mais conhecimentos através de livros e cursos sobre o assunto

Segue aqui o link do Livro Oficial: [Clique aqui](https://www.amazon.com.br/Domain-Driven-Design-Eric-Evans/dp/8550800651?tag=goog0ef-20&smid=A1ZZFT5FULY4LN&ascsubtag=go_726685122_54292137521_242594579893_aud-519888259198:pla-398510643241_c_)

Então vamos ao que interessa...

Arquitetura do Sistema



Agora que você já foi apresentado a imagem que representa a arquitetura do sistema, em seguida segue uma rápida explicação sobre cada camada.

**#1: Camada de Apresentação**

Como o próprio nome sugere, essa é a camada que tem a lógica de **apresentação** para o usuário. É nesta parte que colocamos projetos de ASP.NET, ASP.NET Core, Angular e outros.

A camada de aprensentação tanto pode acessar diretamente o **Application Service** quando for possível quanto também pode acessar o mesmo **Application Service** através da **WEB API**

**#2: WEB API**

Através da WEB API é possível criar classes em C# e publicá-las como serviços web e é normalmente utilizado para conectar seu sistema com outros sistemas feitos em .net mesmo ou em outra tecnologia com Python, Java, node.js, php

E isso é possível porque a WEB API cria serviços em .net que respondem por comandos HTTP (GET, POST, DELETE...)

A WEB API assim como a **camada de apresentação** acessa as partes internas do sistema através

Do **Application Services**

**#3: Application Services**

O Application Services existe na **fronteira entre o mundo externo a camada de domínio** do sistema. Isso que dizer que qualquer consulta externa no domínio, seja pela camada de apresentão ou pela WEB API deve passar pelo Application Services.

Essa camada não é útil somente como um ponto de conexão entre o mundo externo e o domínio. Através dela... quando necessário **podemos mediar regras de negócio entre entidades diferentes** quando for identificado que uma regra não pode ficar vinculada a uma única entidade.

**#4: Camada de Domínio**

Nessa camada que você vai implementar todas classes de negócio assim como os contratos de serviços, contratos de repositório e implementar os serviços de domínio.

Nessa camada.. a tecnologia é o que menos importa. Pense/Foque no que de fato é importante. As Regras do Negócio

**#5: Infraestrutura**

Nessa camada que você vai implementar as classes concretas de repositório e implementar toda a lógica de acesso a base de dados.

**#5: Cross Cutting**

Nesta camada colocamos os componentes que são considerados comuns para a maior parte do sistema como Tratamento de exceções, Logs, Extração para Excel, Controle de Segurança etc e etc.

Agora que você sabe o básico sobre os camadas da arquitetura adotada, vamos para o passo a passo para você aprender como trabalhar nela.

**Passo a Passo**