Visão geral do padrão DDD Domain-Driven Design

COTI Informática
Escola de Nerds

O que é DDD?

- A principal ideia do DDD (Domain Driven Design) é a de que o mais importante em um software não é o seu código, nem sua arquitetura, nem a tecnologia sobre a qual foi desenvolvido, mas sim o problema que o mesmo se propõe a resolver, ou em outras palavras, a regra de negócio.
- Ela é a razão do software existir, por isso deve receber o máximo de tempo e atenção possíveis.
- Em praticamente todos os projetos de software, a complexidade não está localizada nos aspectos técnicos, mas sim no negócio, na atividade que é exercida pelo cliente ou problema que o mesmo possui.

Características do DDD

Alinhamento do código com o negócio

- Contato dos desenvolvedores com os especialistas do domínio é algo essencial quando se faz DDD (o pessoal de métodos ágeis já sabe disso faz tempo)

Favorecer reutilização

- A codificação é voltada para aproveitar um mesmo conceito de domínio ou um mesmo código em vários lugares

Características do DDD

Mínimo de acoplamento

- Com um modelo bem feito, organizado, as várias partes de um sistema interagem sem que haja muita dependência entre módulos ou classes de objetos de conceitos distintos

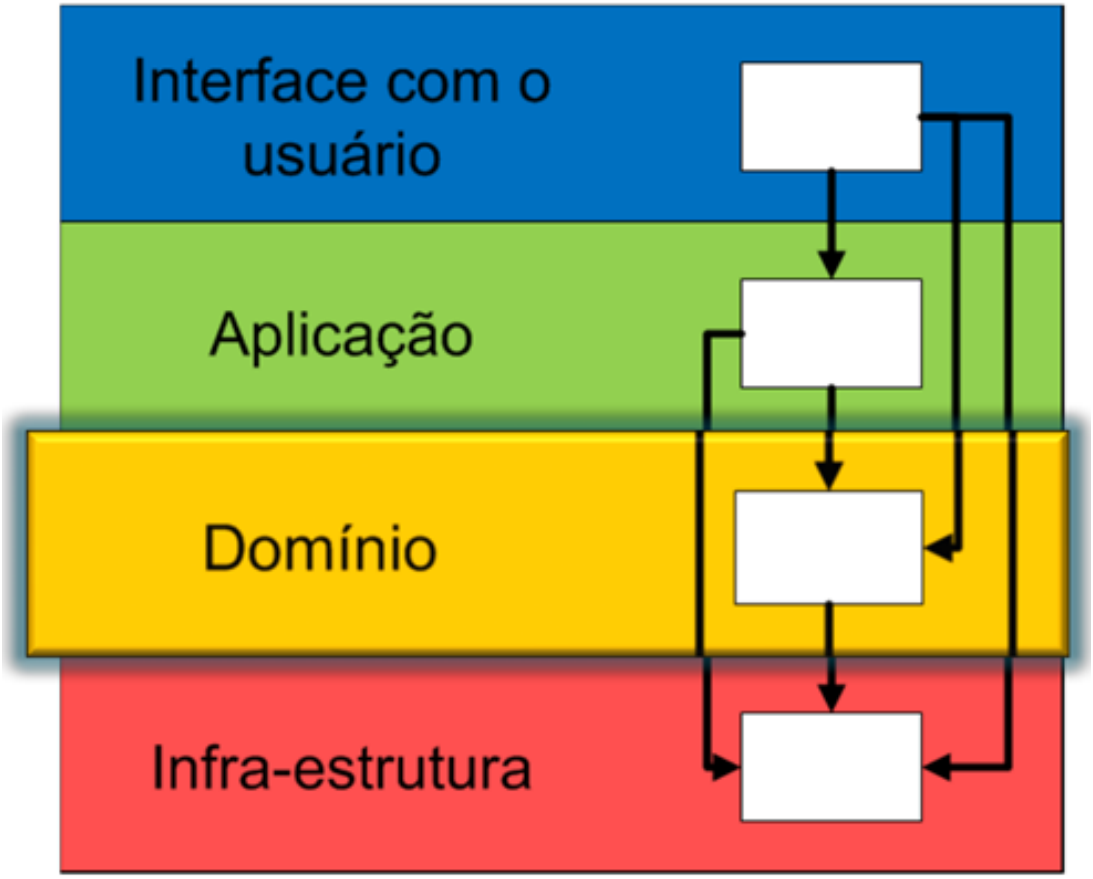
Independência da Tecnologia

- DDD não foca em tecnologia, mas sim em entender as regras de negócio e como elas devem estar refletidas no código e no modelo de domínio. Não que a tecnologia usada não seja importante, mas essa não é uma preocupação de DDD

Construindo uma aplicação baseada em DDD

- Uma vez que decidimos criar um modelo usando MDD, precisamos, inicialmente, isolar o modelo de domínio das demais partes que compõem o sistema.

- Essa separação pode ser feita utilizando-se uma arquitetura em camadas, que dividirá nossa aplicação em quatro partes:



Construindo uma aplicação baseada em DDD

- **Interface de Usuário** – parte responsável pela exibição de informações do sistema ao usuário e também por interpretar comandos do usuário;

- **Aplicação** – essa camada não possui lógica de negócio. Ela é apenas uma camada fina, responsável por conectar a Interface de Usuário às camadas inferiores;

- **Domínio** – representa os conceitos, regras e lógicas de negócio. Todo o foco de DDD está nessa camada. Nosso trabalho, daqui para frente, será aperfeiçoar e compreender profundamente essa parte;

- **Infra-estrutura** – fornece recursos técnicos que darão suporte às camadas superiores. São normalmente as partes de um sistema responsáveis por persistência de dados, conexões com bancos de dados, envio de mensagens por redes, gravação e leitura de discos, etc.

CONCLUSÃO

- Extrair a essência do domínio, dentre milhares de linhas de código de um sistema complexo nem sempre é fácil. O trabalho de refinamento e busca de uma visão clara é contínuo. A refatoração é um processo incessante de busca por melhorias de projeto. Aplicar DDD é utilizar Padrões com o objetivo de extrair e reconhecer a essência de um sistema.