

## Aula 06 Domain-Driven Design (DDD)

### O que significa domínio no contexto de Engenharia de software?

R: No contexto dos sistemas de software, o termo “domínio” refere-se ao conjunto de conhecimentos, regras e conceitos relacionados a uma área específica de negócio ou atividade em que o software será utilizado.

Exemplo: se estivermos desenvolvendo um software para uma empresa de finanças, é necessário ter conhecimento sobre conceitos financeiros, como contabilidade, impostos, investimentos e fluxo de caixa. Se o software for para uma empresa de logística, é importante ter conhecimentos sobre rotas de transporte, tipos de veículos, armazenamento de mercadorias, etc.

### Importância do Domínio

Quando iniciamos um projeto de software, devemos nos concentrar no domínio em que ele operará, pois todo o propósito do software é dar apoio a um domínio específico.

O software precisa incorporar os conceitos e elementos centrais do domínio e implementar com precisão as relações entre eles. Em outras palavras, o software tem que modelar o domínio

Aprendemos muito sobre um domínio enquanto conversamos com seus especialistas. No caso, os especialistas são as pessoas da área que utilizaram o software : bancários, engenheiros, contadores, arquitetos e etc.

**Modelo de Domínio** : é basicamente a representação abstrata de tudo que foi coletado e que será útil na construção do software. Podemos utilizar, conforme uma analogia feita anteriormente, que o “modelo de domínio” (documentado) é uma instância do modelo em si; e por modelo, entende-se os conhecimentos específicos de uma determinada área.

Aprendemos muito sobre um domínio enquanto conversamos com seus especialistas.

O mundo ao nosso redor tem um **excesso** de informações para nossas cabeças lidarem. Mesmo considerando um domínio específico, seus detalhes podem exceder a capacidade da mente humana.

**Por isso, precisamos organizar a informação, sistematizá-la, dividi-la em pedaços menores, agrupar esses pedaços em módulos lógicos.**

*“O sistema bancário certamente deverá conhecer o endereço do cliente, mas não deve se preocupar com a cor dos olhos dele. Esse é um caso óbvio, mas outros exemplos podem não ser tão óbvios”.*

**Representação:** Podemos representar o modelo de domínio de algumas formas.

- Gráficos: Diagramas, imagens, casos de uso, desenho e etc.
- Textos: sentenças que expliquem a visão de domínio.
- Linguagens específica: expressões técnicas e termos do próprio domínio estudado.

*“Para ter o modelo de um domínio, é preciso construir o conhecimento desse domínio”.*

### **Mais sobre o DDD :**

O DDD é uma abordagem que pode ser **usada para auxiliar diferentes modelos de processo de desenvolvimento de software, como o modelo cascata, o Extreme Programming (XP) e o Scrum. O DDD se preocupa predominantemente com as atividades relacionadas à compreensão e modelagem do domínio do negócio em questão.** Isso inclui a identificação dos conceitos do domínio, a definição de limites de contexto, a criação de modelos conceituais e a implementação de código que reflita esses modelos.

**DDD e Linguagem Obíqua :** Um princípio básico do DDD é construir uma linguagem que reflita o modelo, já que o modelo é o local onde o software encontra o domínio. Essa linguagem é chamada de linguagem ubíqua

Linguagem obíqua é um termo usado no contexto do Domain Driven Design (DDD) para se referir a uma linguagem compartilhada e precisa entre todos os membros de uma equipe de desenvolvimento de software e stakeholders do projeto. Essa linguagem deve ser específica ao domínio do problema sendo resolvido e permitir que as partes

envolvidas comuniquem-se claramente e com eficácia. A linguagem obíqua ajuda a garantir que as expectativas e requisitos do negócio sejam entendidos e implementados corretamente no software desenvolvido.