

Resenha – Domain-Driven Design Reference (Capítulos 4, 5 e 6)

Nos capítulos 4, 5 e 6 do livro *Domain-Driven Design Reference*, Eric Evans aprofunda a discussão sobre como modelar domínios de forma clara e sustentável, trazendo conceitos fundamentais para quem deseja aplicar DDD na prática. O texto segue a linha direta e objetiva do autor, mesmo assim consegue mostrar o impacto que essas ideias têm no desenvolvimento de software real.

O capítulo 4 fala sobre **Entidades (Entities)**, que são objetos definidos principalmente por sua identidade, e não apenas pelos atributos que possuem. Evans explica que, em um sistema, duas entidades podem ter exatamente os mesmos valores de propriedades, mas ainda assim serem diferentes se tiverem identidades distintas. Essa noção é essencial quando se trabalha com dados que mudam ao longo do tempo, como usuários, pedidos ou contas bancárias. O autor também destaca a importância de manter as entidades pequenas e coesas, evitando que elas se tornem estruturas pesadas e difíceis de manter.

No capítulo 5, o foco está nos **Objetos de Valor (Value Objects)**. Diferente das entidades, eles não possuem identidade própria, mas são definidos apenas pelos valores que carregam. São imutáveis por natureza e podem ser substituídos completamente quando sofrem alterações. Evans mostra que tratar valores como objetos enriquece o modelo, já que isso permite encapsular comportamentos junto com os dados. Um exemplo simples é representar dinheiro ou uma medida de distância como objetos de valor, em vez de apenas usar números soltos. Essa prática ajuda a dar mais clareza ao código e a evitar erros comuns.

Já o capítulo 6 apresenta o conceito de **Serviços (Services)**, que representam ações ou operações importantes do domínio que não se encaixam naturalmente em entidades ou objetos de valor. Evans observa que nem toda lógica cabe dentro de um objeto, e forçar isso pode tornar o modelo confuso. Os serviços permitem expressar essas funcionalidades de forma clara e explícita, geralmente usando nomes que refletem operações do negócio. Assim, eles ajudam a manter o modelo mais limpo e coerente.

A leitura desses três capítulos mostra como DDD se preocupa em organizar o software de acordo com a forma como o domínio funciona, e não apenas pela lógica técnica. Entidades, objetos de valor e serviços são blocos fundamentais que, juntos, tornam o modelo mais expressivo, fácil de entender e alinhado às regras de negócio.

Em resumo, Evans apresenta nesses capítulos princípios que parecem simples, mas que têm grande impacto na prática. Ele mostra que projetar bem não significa complicar, mas sim encontrar a melhor forma de refletir a realidade do domínio dentro

do software. Esses conceitos ajudam a criar sistemas mais robustos, flexíveis e próximos da linguagem que os especialistas do negócio realmente usam no dia a dia.