

Engenharia de Softwares Escaláveis

Design Patterns e Domain-Driven Design com Java

Agenda

Etapas 3: Introdução ao Domain-Driven Design.

- Tipos de Empresas e Sistemas.
- Projeto Tático.



Tipos de Empresas e Sistemas

Domain-Driven Design não é apenas uma questão técnica, mas sobretudo profundamente estratégica.

A posição da TI na organização e a natureza dos sistemas que desenvolvemos afetam decisões arquiteturais, de investimento e até mesmo das carreiras dos profissionais envolvidos.



Tipo de Empresa	Papel da TI	Exemplos	Reflexões
TI como atividade-fim	A TI é o produto principal da empresa.	Nubank, iFood, Netflix, Spotify, empresas de SaaS, startups tech.	<p>Desenvolver software é o <i>core business</i>.</p> <p>TI é protagonista e precisa inovar constantemente.</p> <p>O código é o produto.</p>
TI como atividade-meio	A TI apoia a atividade principal (não é o core).	McDonald's, Petrobras, Bancões, Hospitais, Seguradoras.	<p>TI é essencial, mas não é o foco da empresa.</p> <p>Software precisa ser confiável e manter o negócio rodando, mas inovação radical <u>nem sempre</u> é prioridade.</p>

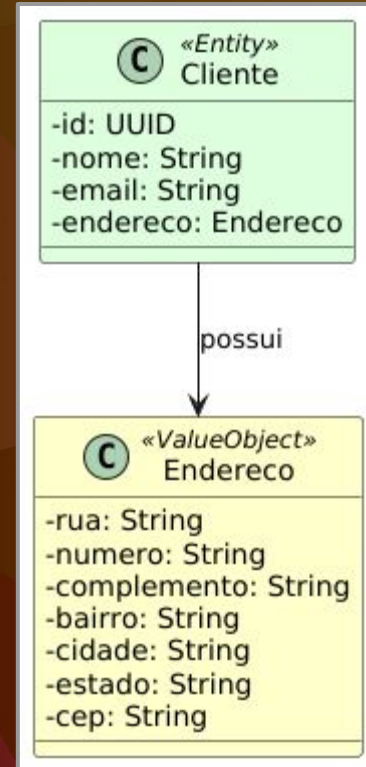
Tipo de Domínio	Descrição	Exemplo em uma empresa	Papel da equipe de desenvolvimento
Core Domain	<p>É onde as empresas realmente competem.</p> <p>Contém a inteligência de negócio que diferencia a empresa no mercado.</p>	Algoritmo de recomendação da Netflix, Sistema de crédito do Nubank, Atendimento do McDonald's.	A equipe é estratégica, precisa entender profundamente o domínio e trabalhar próxima aos especialistas.
Subdomínio de Suporte	<p>Dá apoio ao core, mas não é exclusivo.</p> <p>Pode ter peculiaridades.</p>	Gestão de RH, faturamento, compliance, logística interna.	<p>Pode ter complexidade, mas não é diferencial competitivo.</p> <p>Pode ser parcialmente terceirizado ou reutilizar frameworks.</p>
Subdomínio Genérico	Problemas amplamente resolvidos e não específicos do domínio da empresa.	Autenticação de usuários, envio de e-mails, monitoramento de logs, Folha de Pagamento, Contabilidade.	<p>Evita-se reinventar a roda.</p> <p>Usa-se bibliotecas, SaaS, soluções de mercado.</p> <p>Ideal para ser comprado ou plugado.</p>



Projeto Tático

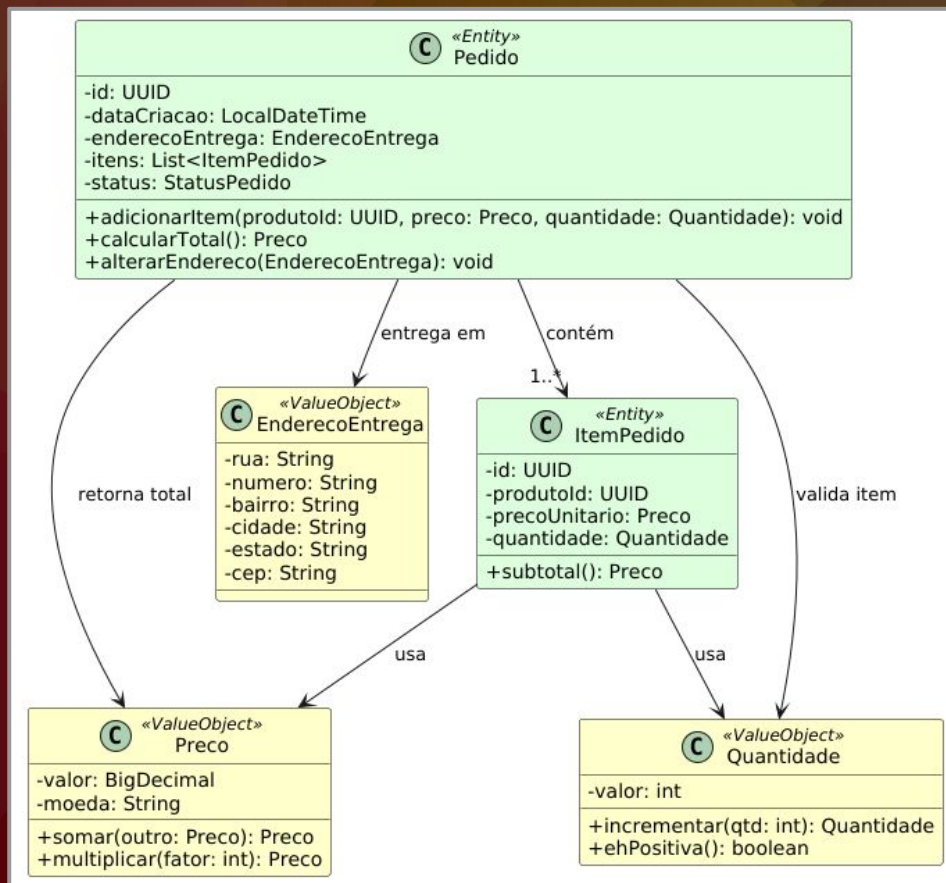
É importante identificar e modelar as **Entidades** e **Objetos de Valor** que representam os conceitos fundamentais do **domínio**.

Entidades são objetos identificáveis com uma **identidade única** e que mudam ao longo do tempo, enquanto **Objetos de Valor** são **objetos imutáveis** que representam valores específicos dentro do domínio.



Agregados são grupos de entidades e objetos de valor que são tratados como uma unidade coesa dentro do **modelo de domínio**.

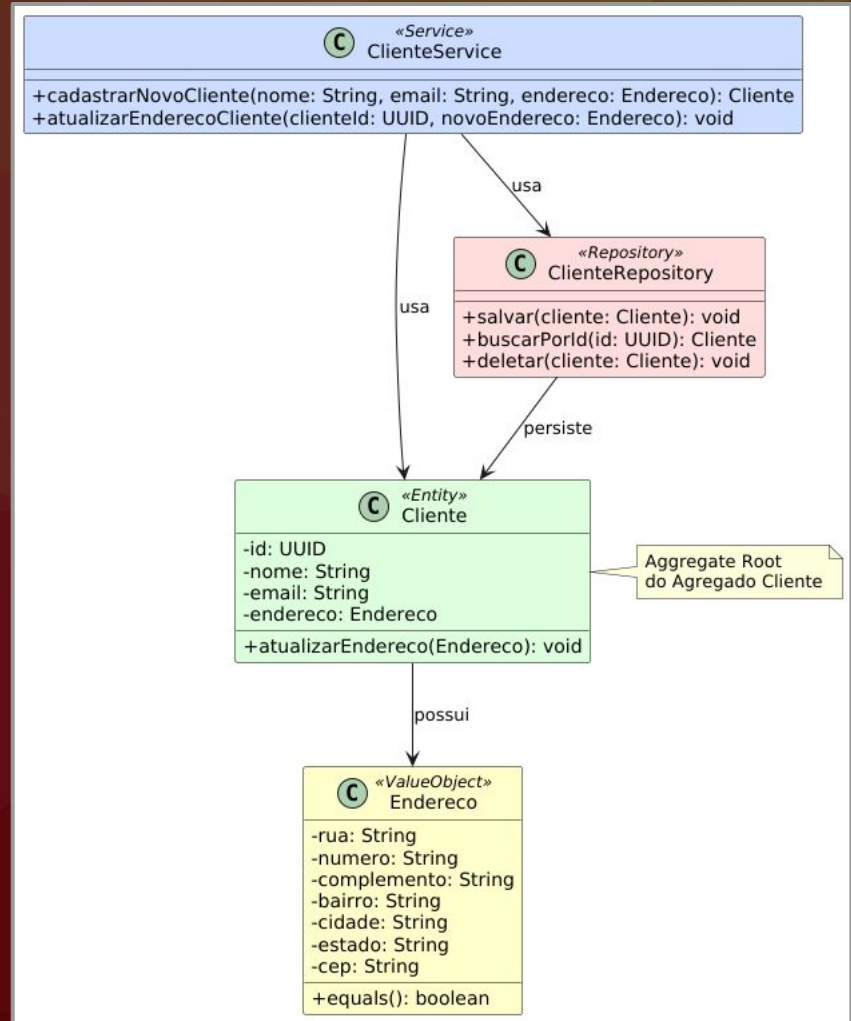
Projetar software de forma tática no **DDD** envolve a definição e implementação de **Agregados** para garantir a consistência e a integridade dos dados dentro do sistema.



Elemento	Tipo	Papel
Pedido	Entidade	Raiz do agregado; identidade própria, com invariantes (ex: status, total)
ItemPedido	Entidade	Parte do agregado, com identidade (pode ser composta)
EnderecoEntrega	Value Object	Descreve onde entregar; não possui identidade
Preco	Value Object	Evita lógica monetária dispersa; encapsula operações
Quantidade	Value Object	Encapsula lógica e validações de quantidades

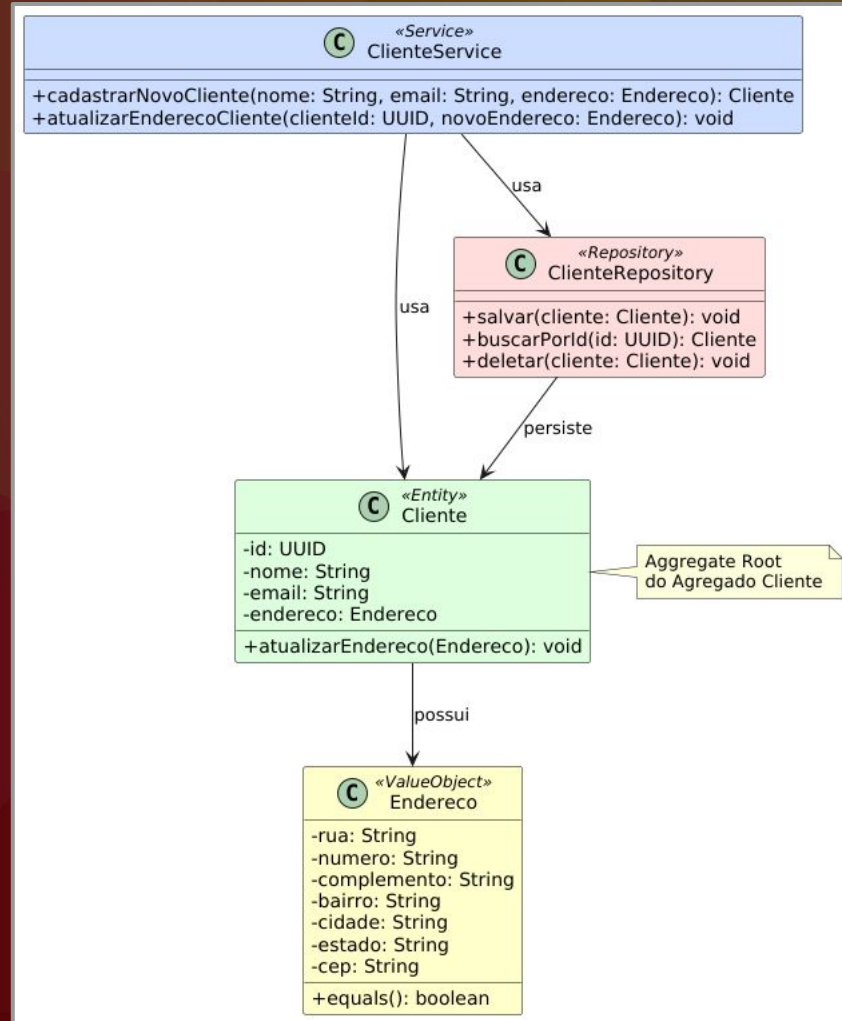
Repositórios são responsáveis por abstrair o acesso aos dados subjacentes e fornecer uma interface para recuperar e persistir objetos de domínio.

No contexto tático do **DDD**, é importante projetar e implementar repositórios que se alinhem com os conceitos do modelo de domínio e que forneçam uma interface consistente e eficaz para interagir com os dados.



Serviços de domínio encapsulam a lógica de negócio complexa que não se encaixa naturalmente em **entidades** ou **objetos de valor** individuais.

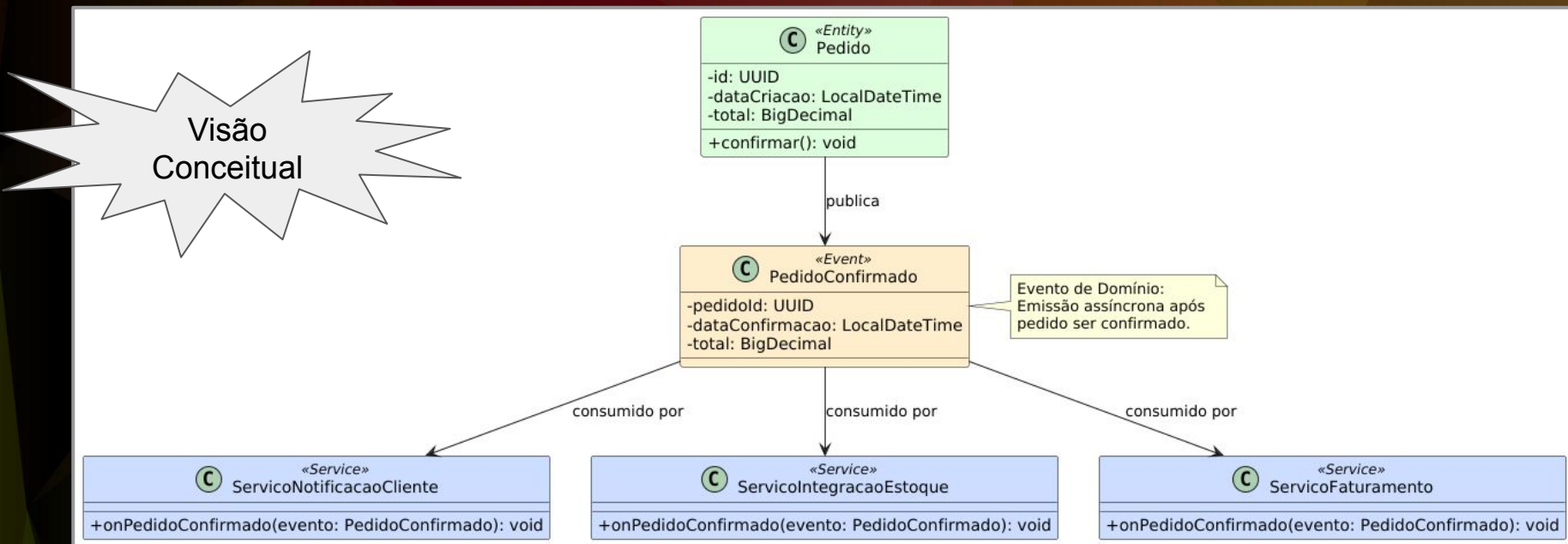
Projetar software de forma tática no **DDD** envolve a identificação e implementação de serviços de domínio que representam as operações fundamentais do negócio e promovem a reutilização e a coesão do código.



Elemento	Tipo	Função no DDD
Cliente	Entidade	Possui identidade e encapsula regras de negócio
Endereco	Value Object	Sem identidade, imutável, definido apenas por seus atributos
ClienteService	Domain Service	Contém lógica de negócio que não pertence a uma única entidade
ClienteRepository	Repository	Abstrai o acesso a dados do agregado, como se fosse uma coleção em memória

Eventos de Domínio são **eventos** significativos que ocorrem dentro do contexto do domínio e que podem ser capturados e processados pelo sistema.

Projetar software de forma tática no **DDD** envolve a modelagem e a implementação de eventos de domínio que representam as **mudanças de estado importantes** dentro do sistema e que podem ser usados para comunicar informações entre diferentes partes do sistema.



Visão de Projeto

