Introdução ao Domain-Driven-Design DDD

Ivan Paulovich
Arquiteto de Softwares

Banco Olé Consignado – Julho/2017

Ivan Paulovich

Microsoft Most Valuable Professional

De 2012 à 2014

100LOOP

Microsoft Most Valuable Professional Reconnect

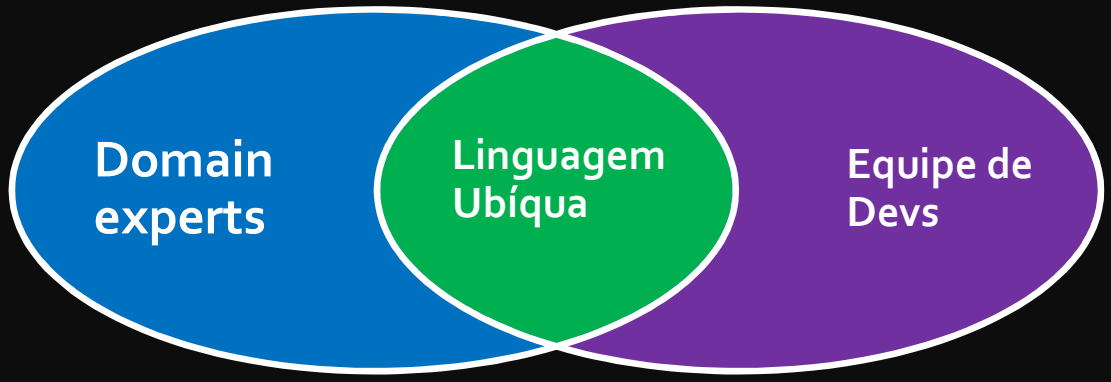
bhs

.net RAPTORS

Agenda

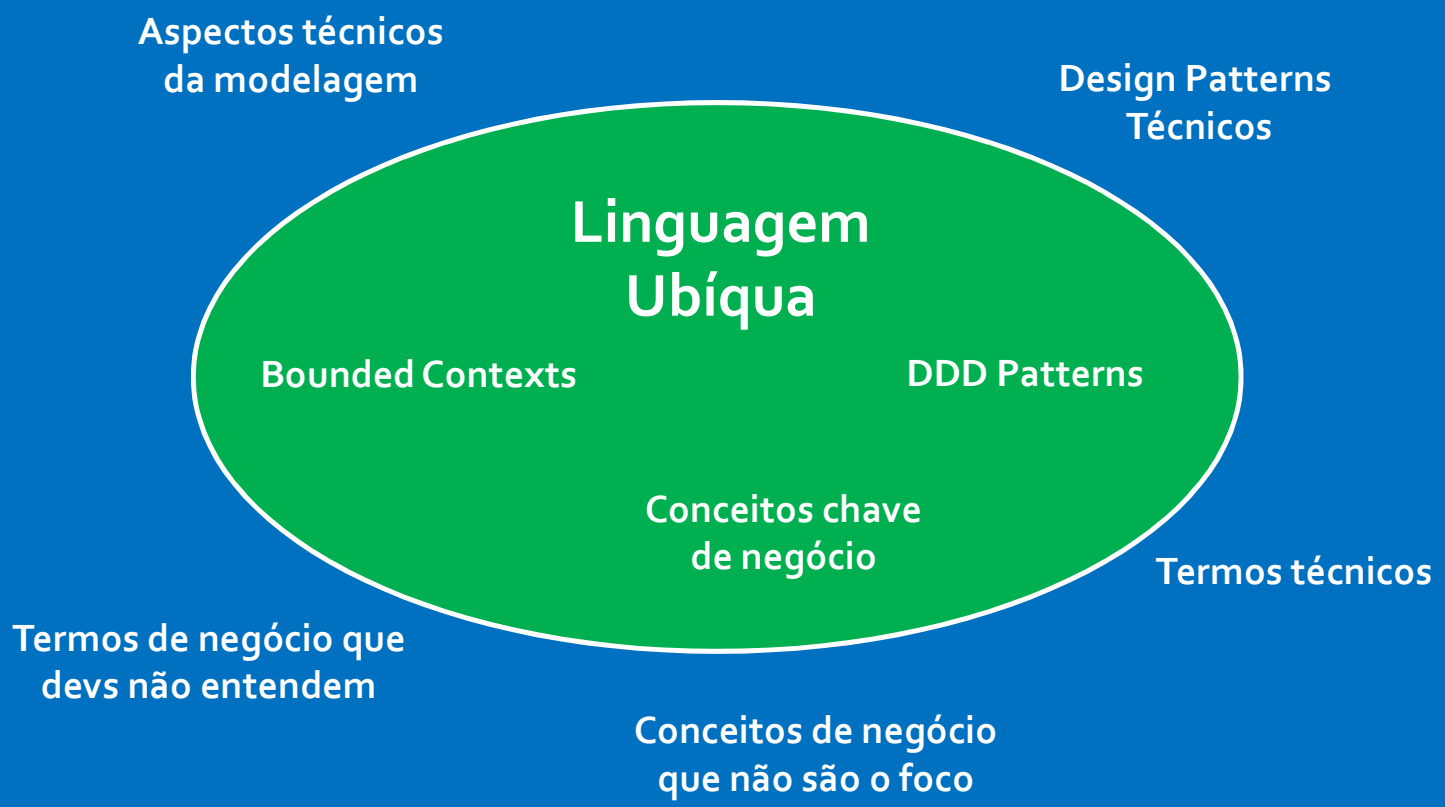
- Linguagem Ubíqua
- Bounded Context
- Tipos de Domínio
- Core Domain
- Livros sobre DDD
- Estilos Arquiteturais
- Building Blocks
- Dúvidas?

Linguagem Ubíqua



Palavras e verbos que refletem a **semântica do negócio** restritos a um **contexto.**

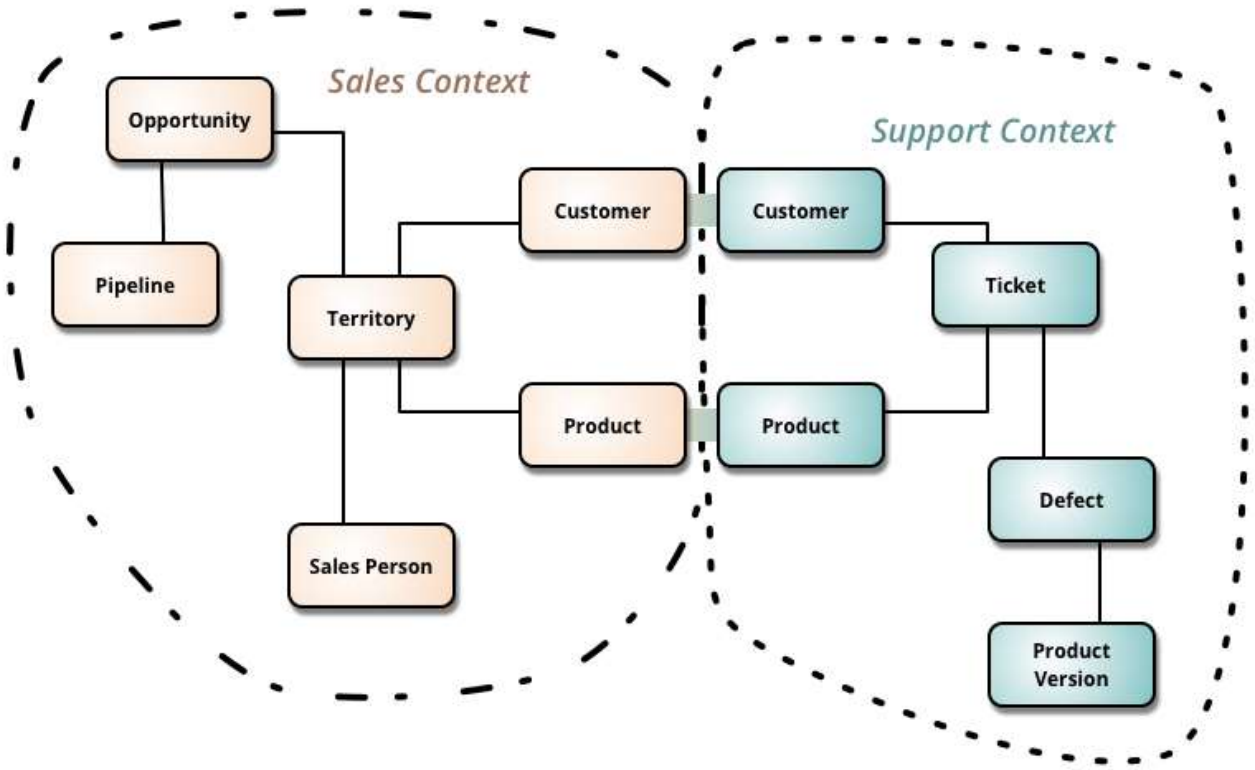
Linguagem Ubíqua



Linguagem Ubíqua

- A linguagem ubíqua é falada dentro de um Bounded Context.
- Os termos precisam ser claramente definidos, não ambíguos e consistentes.
- Muito importante numa conversação entre especialistas de domínio e desenvolvedores.
- A linguagem ubíqua deve evoluir progressivamente.
- Se a linguagem ubíqua não está clara então há trabalho a ser feito.

Bounded Context

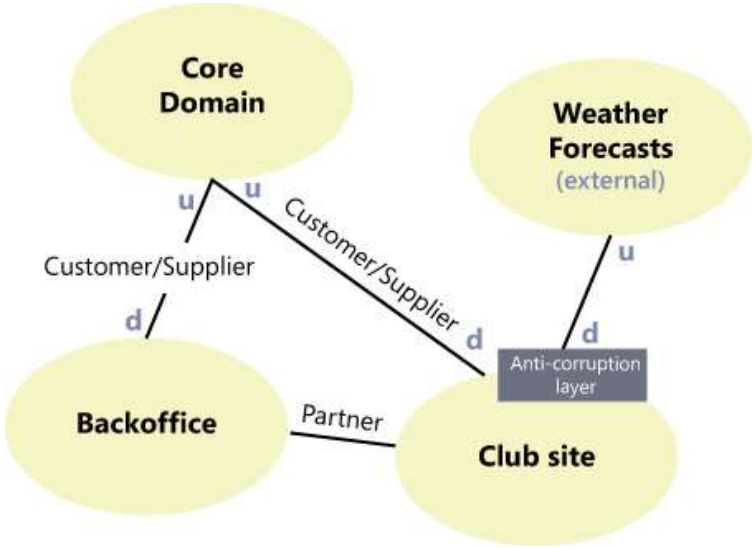


https://martinfowler.com/bliki/BoundedContext.html

Bounded Context

- Áreas de domínio exploradas isoladamente
- Definido ao longo que os requisitos são avaliados e a linguagem construída

Context Map



Relacionamentos

- Partnership
- Shared Kernel
- Open Host Service
- Customer/Supplier
- Big Ball Of Mud
- Conformist
- Anticorruption Layer
- Separate Ways
- Published Language

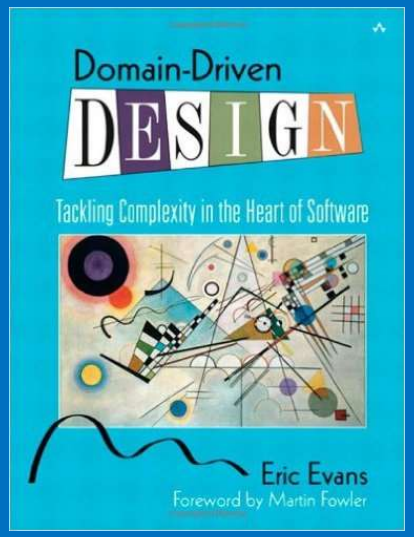
Tipos de Domínio

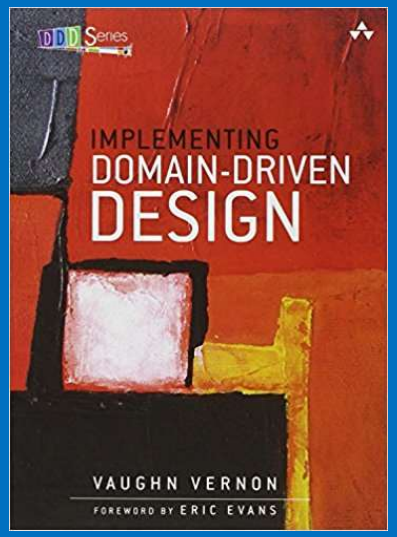
- Um Modelo pode ser:
  - O Core Domain
  - Um Domínio de Suporte
  - Um Domínio Genérico
- **Foque o esforço de modelagem no Core Domain**
- Considere outsourcing pra o Domínio de Suporte
- Adquira Domínios Genéricos

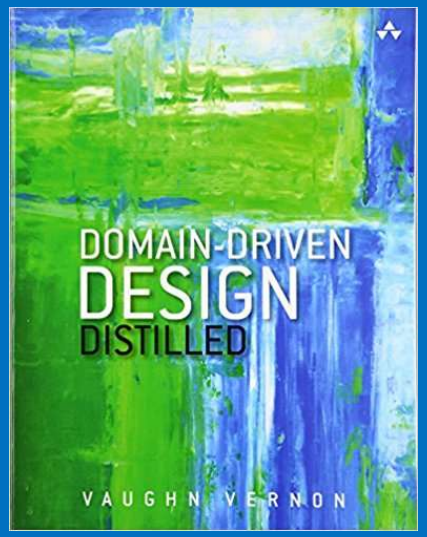
Core Domain

- Identifique o seu Core Domain
- Analise o seu Core Domain
- Foque os recursos no Core Domain

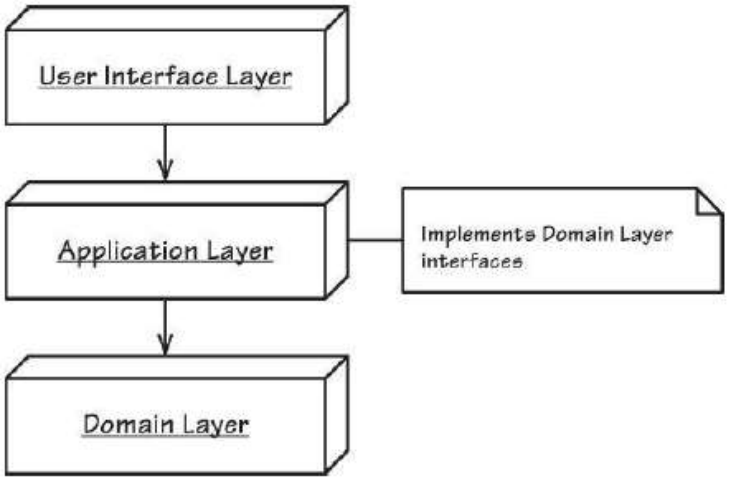
Livros sobre DDD



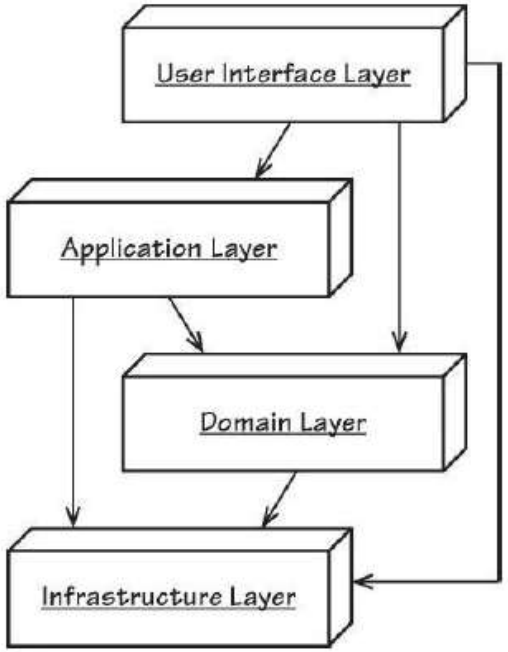




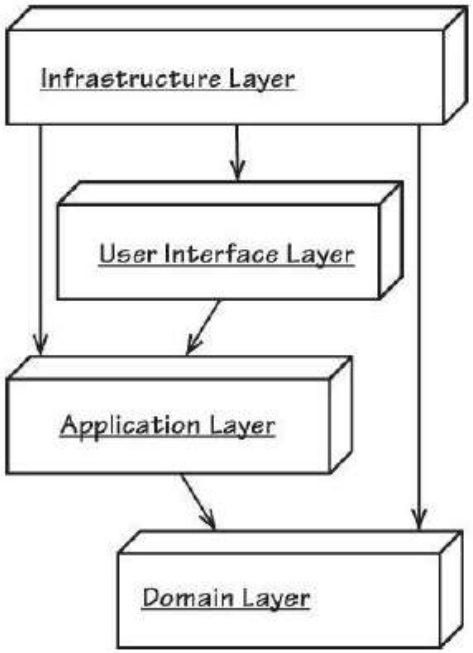
Estilos Arquiteturais



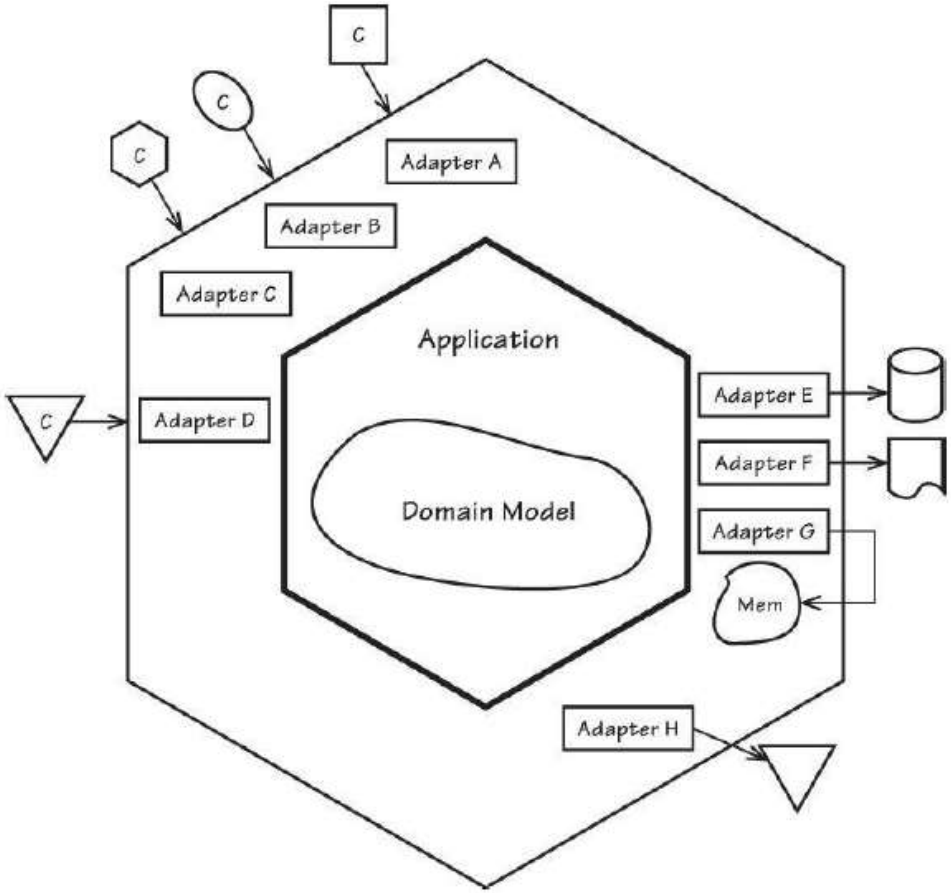
Estilos Arquiteturais



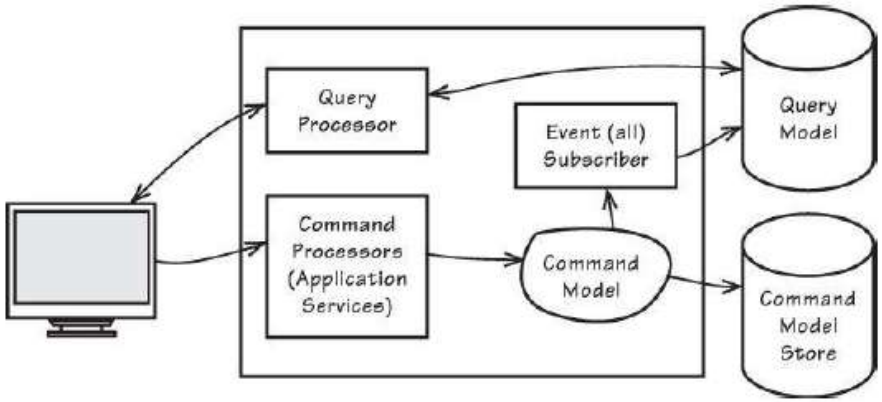
Estilos Arquiteturais



Estilos Arquiteturais



Estilos Arquiteturais



Dependency Inversion Principle

"Módulos de mais alto nível não devem depender de módulos de mais baixo nível. Ambos devem depender de abstrações.

Abstrações não devem depender de detalhes. Detalhes devem depender de abstrações."

Robert C. Martin

DDD Building Blocks

* Entidades
* Repositórios
* Tipos de Valor
* Serviços
* Agregações
* Domain Events

Entidades

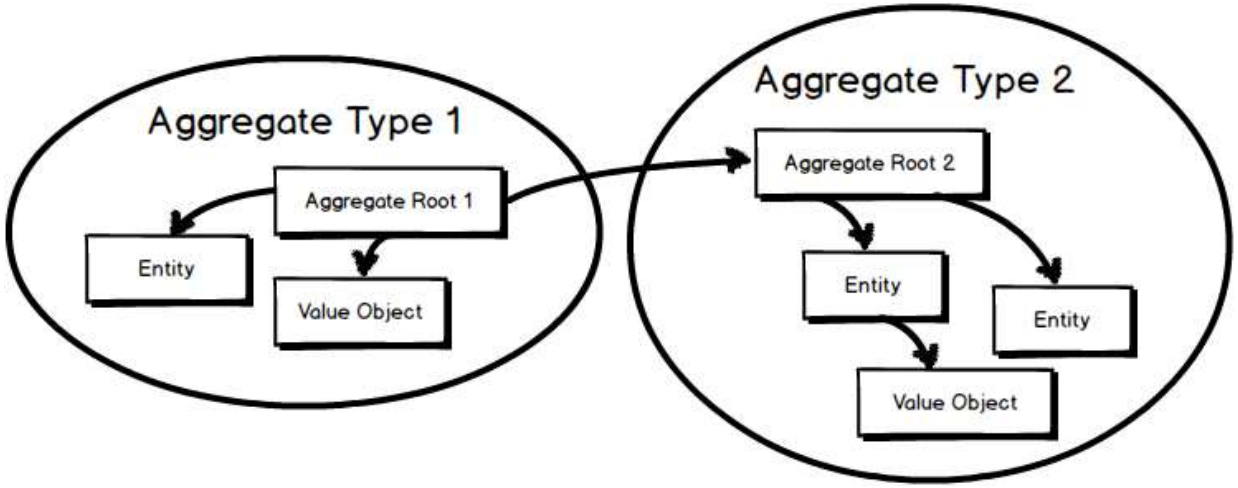
- Possuem uma Identidade e Estado
- Persistência
- Somente uma responsabilidade
- Pode ser constituída por outras entidades e tipos de valor

Tipos de Valor

- Não possuem identidade
- Imutável
- Definido pelos seus atributos
- Regras de negócio fazem parte dos tipos de valor

Agregações

- Um grupo de entidades e tipos de valor relacionados
- Definem um escopo de transação e de tratamento de concorrência
- Um Bounded Context terá múltiplas agregações



Agregação Raiz

- Uma entidade pode ser uma agregação raiz
- Referências externas somente conhecem agregações raiz
- Persista a agregação raiz e o grafo de objetos relacionados

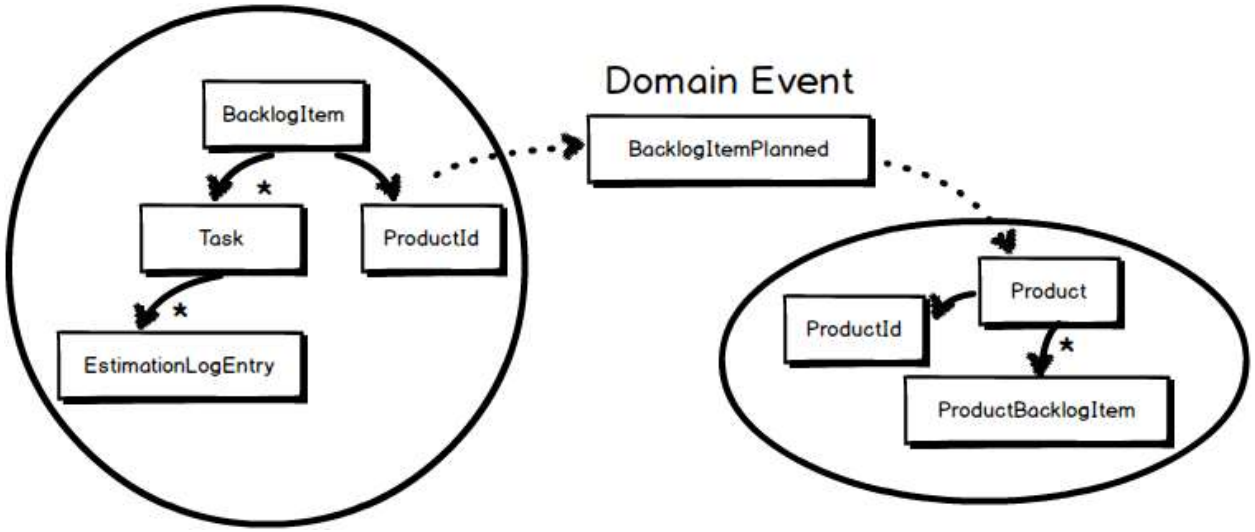
Repositórios

- Realizam a persistência de uma agregação

Serviços

- Regras e operações que não pertencem naturalmente a um objeto de domínio
- Não guardam estado
- São imutáveis

Domain Events



Dúvidas?

Referências

- Introduction to DDD - Adam Štipák
  https://www.slideshare.net/WEBtlak/introduction-to-ddd-adam-tipk
- Domain-Driven Design at ZendCon 2012
  http://bradley-holt.com/2012/11/domain-driven-design-at-zendcon-2012/
- Architecting and Implementing Domain-Driven Design Patterns with Microsoft .NET
  https://channel9.msdn.com/Events/TechEd/Europe/2014/DEV-B211
- Modern Software Architecture-Domain Models, CQRS, and Event Sourcing – Notes
  https://www.slideshare.net/ChinhNguyen49/modern-software-architecturedomain-models-cqrs-and-event-sourcing-notes
- Domain Driven Design
  https://pt.slideshare.net/shadrik/domain-driven-design-52410778
- **Modeling Aggregates with DDD and Entity Framework**
  https://vaughnvernon.co/?p=879