

EMBARCADERO CONFERENCE



 embarcadero®

Desmistificando Domain-Driven Design(DDD)

Gustavo Mena Barreto



CONFERENCE

EMBARCADERO CONFERENCE 2022

Agenda

DDD

Vantagens

Principais
Conceitos

Exemplo
Delphi

Dicas Finais

Dúvidas
Contatos

Um pouco sobre mim

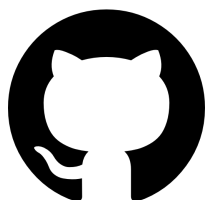
Gustavo Mena Barreto

Analista de Sistemas na Aquasoft.

Formado em Análise e Desenvolvimento de Sistemas pela Unisinos/RS.

Mais de 10 anos de experiência com Delphi.

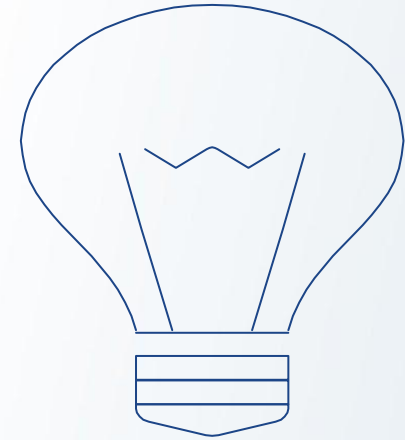
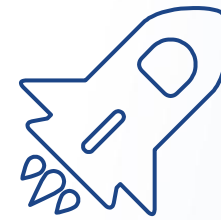
Palestrante Embarcadero Conference 2021 e DelphiCon 2021.



<https://github.com/gustavomenabarreto/Delphi-DDD>



DDD???

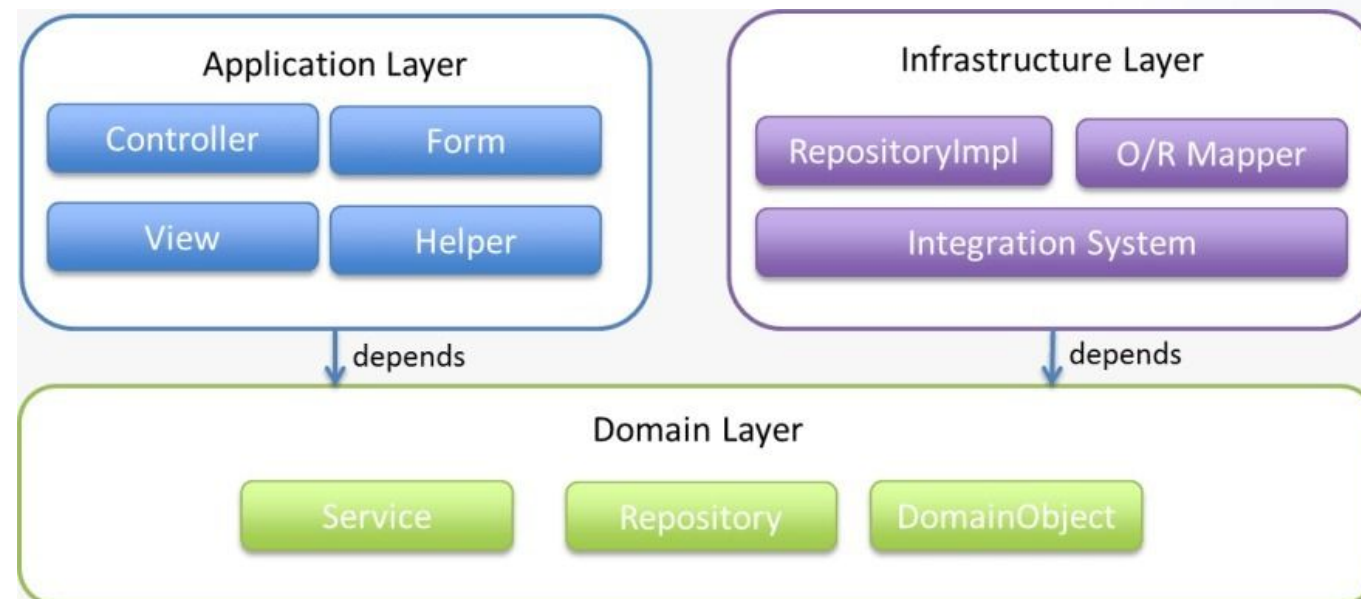


DDD - Domain Driven Design

- Design guiado pelo domínio
- Modelagem de software focada em resolver problemas da complexidade do negócio

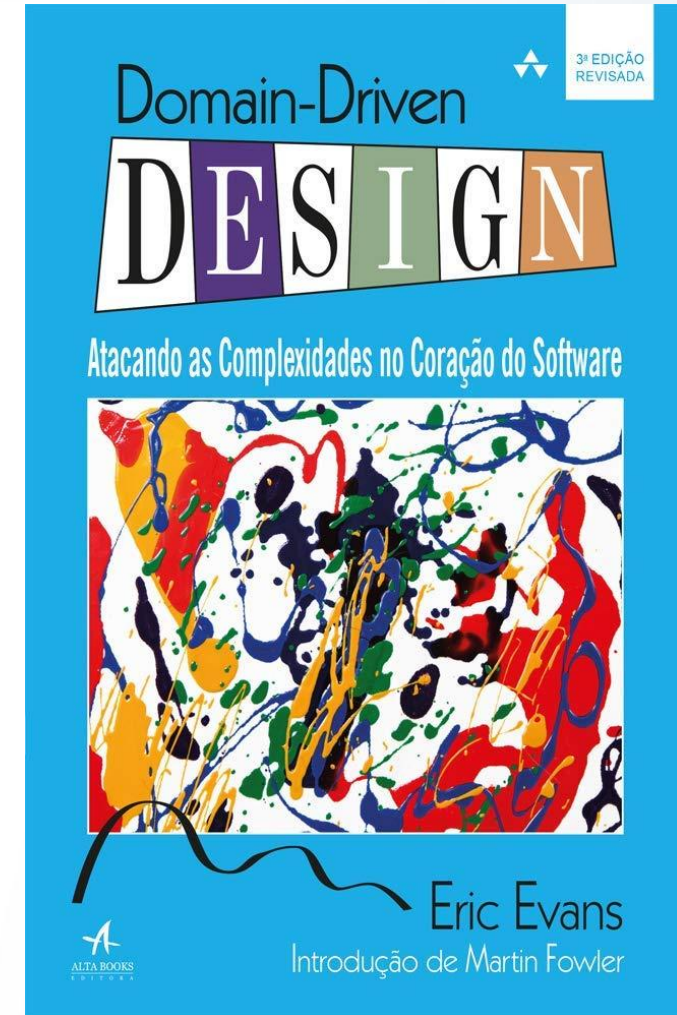
DDD - Domain Driven Design

- Não é uma tecnologia
- Não é uma arquitetura em camadas

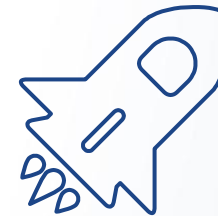


DDD - Domain Driven Design

**Domain-Driven Design:
Atacando as
complexidades no
coração do software -
Eric Evans**



VANTAGENS



Vantagens

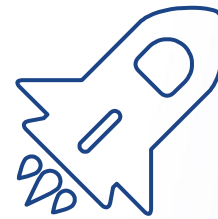
A lógica do domínio
totalmente independente
de qualquer tecnologia
para a implementação

Código mais limpo

Alinhamento entre
negócio e
desenvolvimento



Principais Conceitos



- DDD
- Domain
- Context Map
- Entity
-
-

Papéis

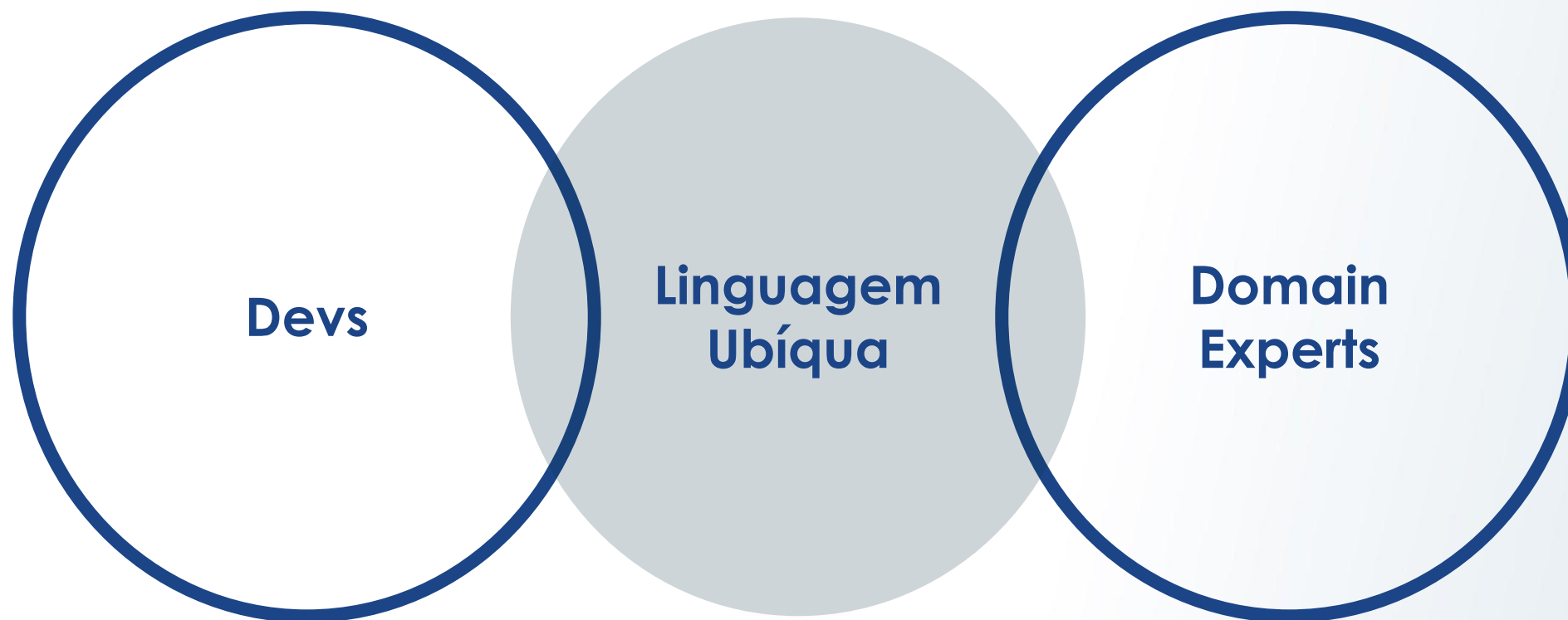
Time de desenvolvimento

**Especialistas no negócio
(Domain Experts)**

Termos Técnicos

Termos das regras de
negócio

Devs X Domain Experts



Linguagem Ubíqua

Utiliza as terminologias da realidade do negócio

Definida entre os desenvolvedores e os experts no negócio

Objetivo de evitar erros de comunicação

Domínio e Modelo

Domínio

- Coração do negócio
- Razão do negócio e do desenvolvimento do software existir

Modelo

- Abstrai a complexidade do negócio
- Meio de comunicação usado pelo time
- Evolutivo
- Especificação = Implementação

Bounded Context e Context Map

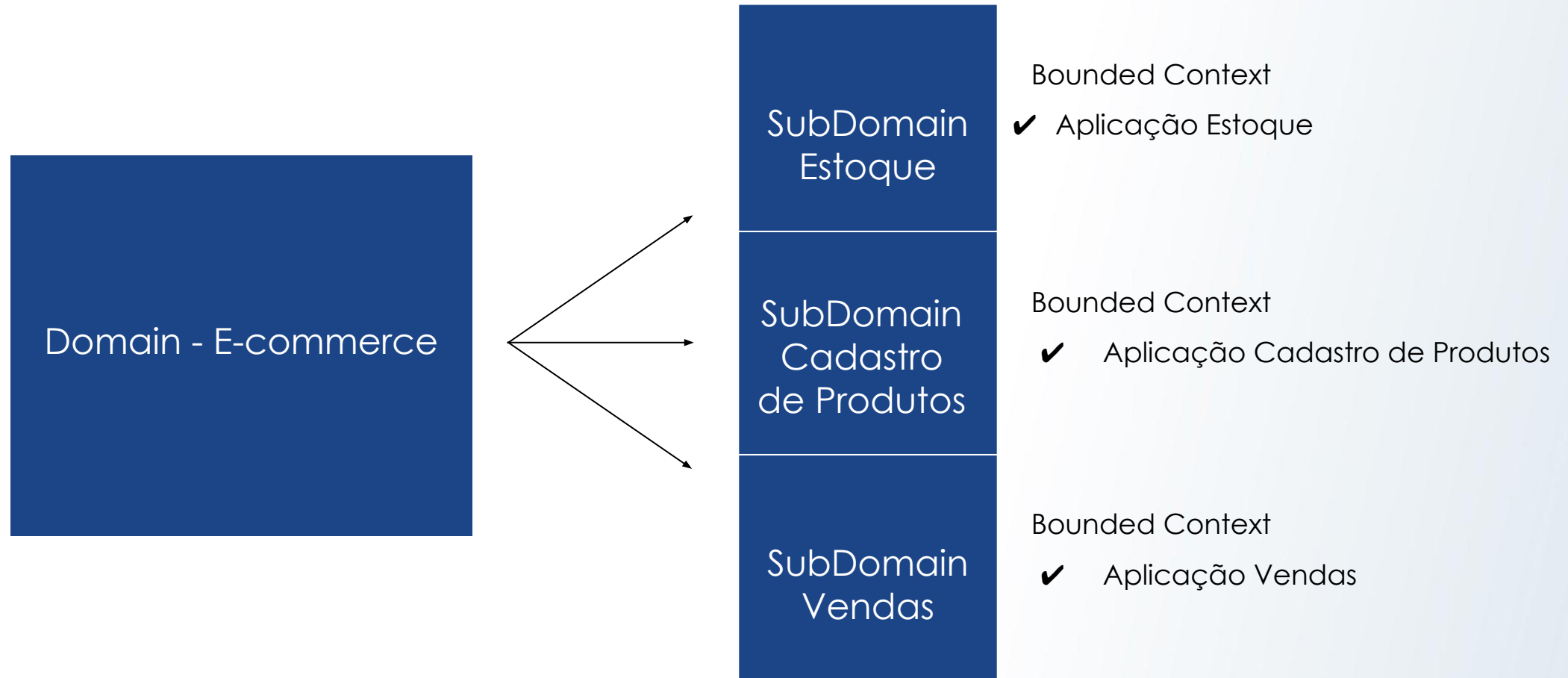
Bounded Context

- Utilizado para múltiplos Subdomínios/modelos
- Define o contexto e aplicabilidade do Subdomínios/modelos e como se relaciona com outros Subdomínios/modelos

Context Map

- Modelo Visão Geral do Software
- Mapeamento de todos os Bounded Context e os relacionamentos entre eles

Exemplo



Representando o Modelo

Entity

Value Objects

Repositories

Services

Factory

Entidade - Entity

- Não são definidos por seus atributos
- Possuem identificador exclusivo
- Potencialmente mutável
- Possui todo o histórico de transição

Objetos de valor - Value Objects

- Um objeto que agrega valor às entidades, não possui identidade e é imutável
- Contém os atributos da entidade
- São definidos pelo valor dos atributos

Repositorios - Repositories

- Acesso a camada de dados
- Realiza a persistência e consultas dos dados das entidades

Serviços - Services

- Implementam a lógica de negócio
- Trabalha com diversas entidades e realiza persistência através de repositórios

Factory

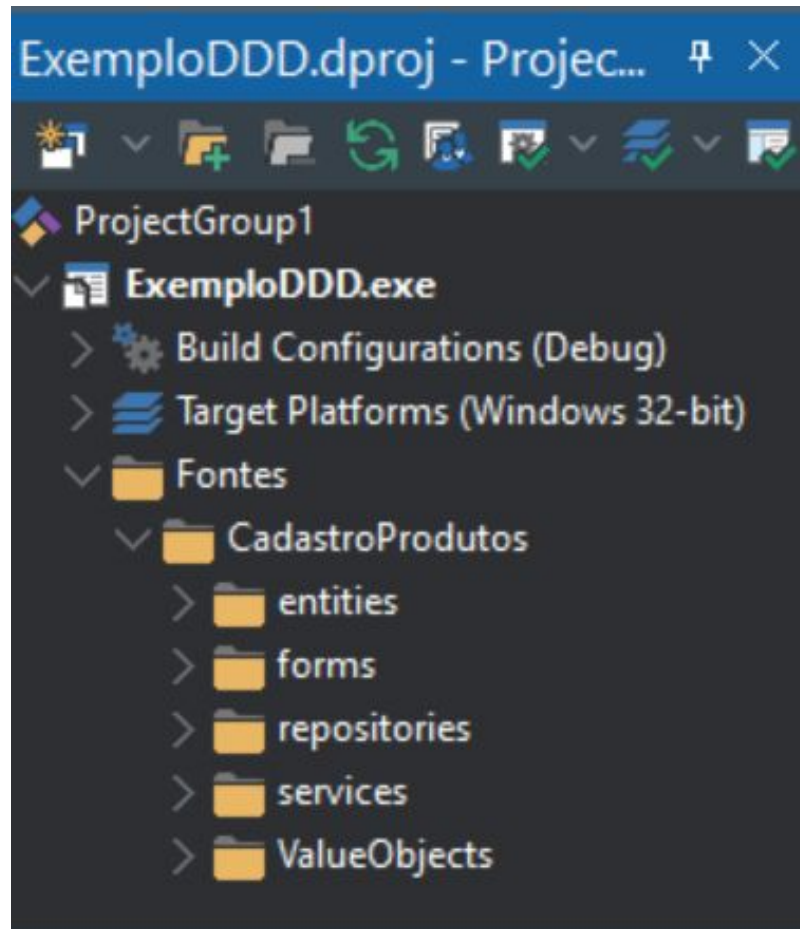
- Responsável por construir adequadamente um objeto/entidade

Hora do código

EXEMPLO EM DELPHI



Estrutura



Estrutura do projeto
utilizando os
principais conceitos

DICAS FINAIS



Dicas Finais

Leia livros e conteúdos antes de se aventurar profissionalmente

Vai ser desafiador.

Experimente muito e espere cometer muitos erros

Não desanime, a modelagem é um processo criativo.

É mais adequado implementar apenas em projetos de sistemas complexos

A jornada vai valer a pena no final

Avaliação Palestra

Resgate de Brindes



Dúvidas?

 <https://br.linkedin.com/in/gustavo-mena-barreto-59670512a>

 gustavo.barreto@aquasoft.com.br

OBRIIGADO