

TRILHA ARQUITETURA



Arquitetura Hexagonal e DDD

André Racz

Diretor de Arquitetura na Avanade

<https://www.linkedin.com/in/aracz/>



THE
DEVELOPER'S
CONFERENCE

Perguntas e Respostas / Interação

- Acesse vevox.com
- ID: 156-880-660



Conhecimento sobre Arquitetura Hexagonal



- Grau de conhecimento sobre Arquitetura Hexagonal:
 - Nenhum
 - Já li / vi palestra a respeito
 - Já fiz uma prova de conceito
 - Já usei profissionalmente



Acesse vevox.app
ID: 156-880-660



THE
DEVELOPER'S
CONFERENCE

Arquitetura Hexagonal

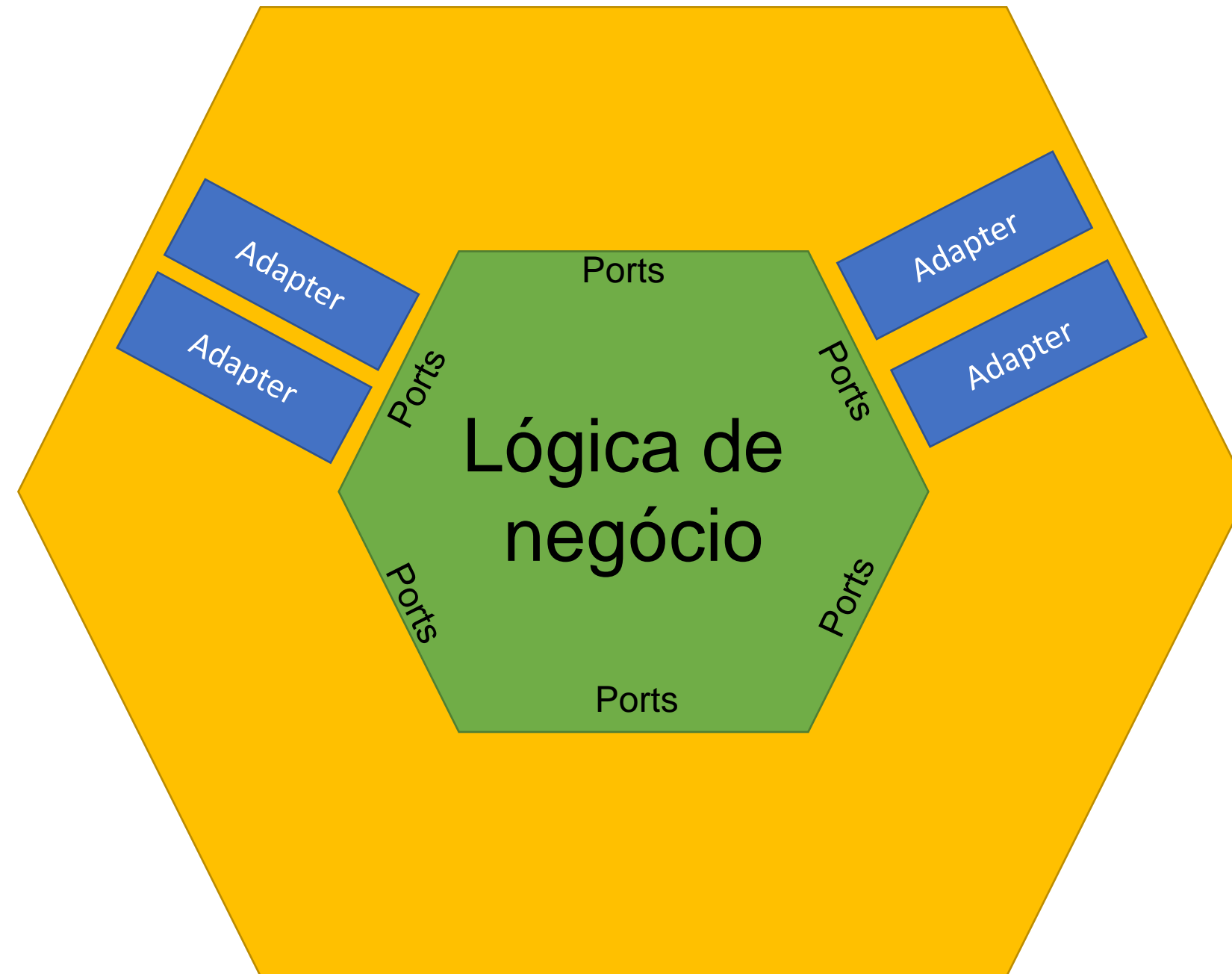
- Criada em 2005 por Alistair Cockburn
- Também chamada de Port & Adapters
- <https://alistair.cockburn.us/hexagonal-architecture/>

“Crie uma aplicação para trabalhar sem uma UI ou um banco de dados para que você possa rodar testes automatizados de regressão na aplicação, trabalhar quando o banco de dados ficar indisponível e ligar aplicações sem envolvimento de usuários.”

Arquitetura Hexagonal



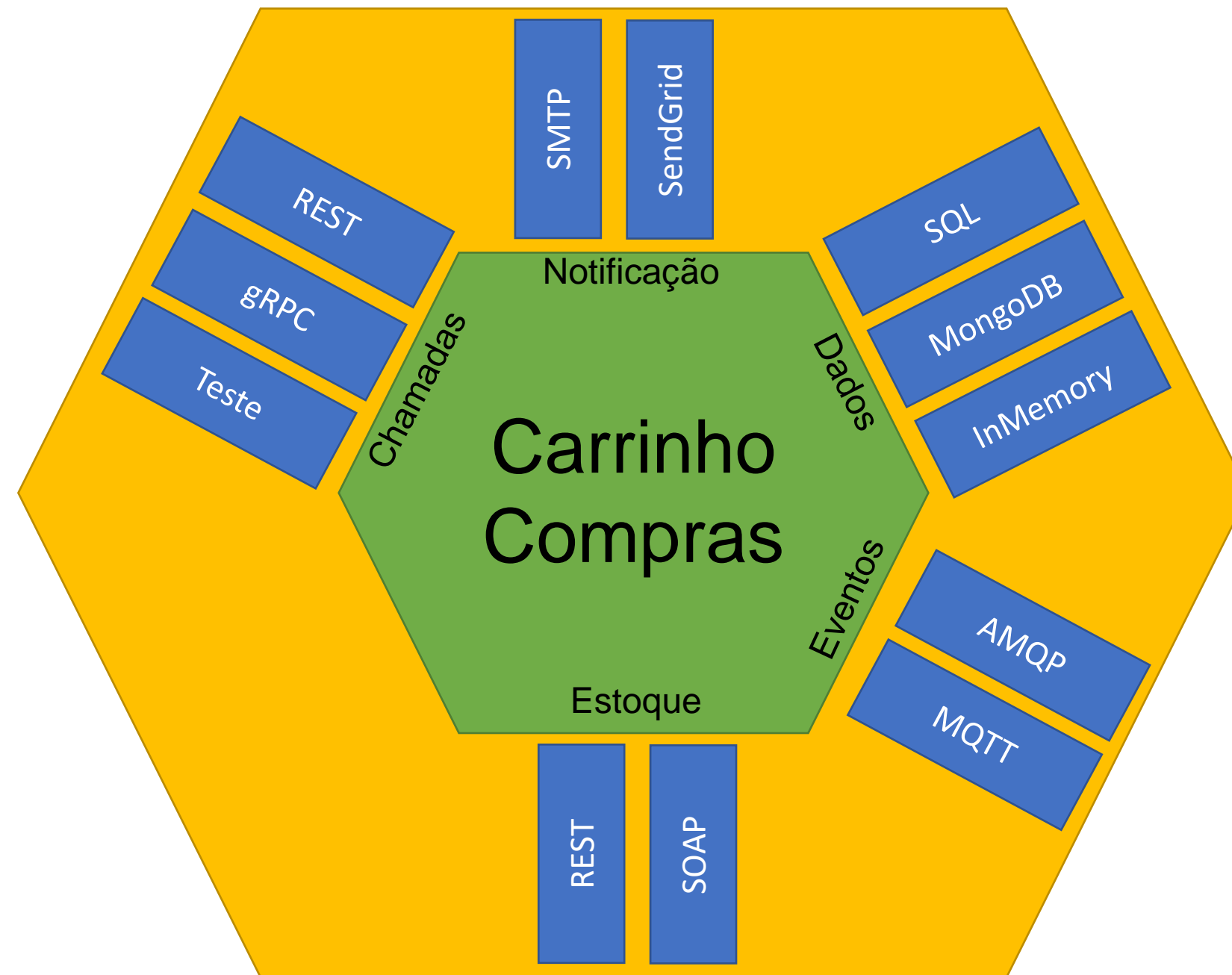
THE
DEVELOPER'S
CONFERENCE



Arquitetura Hexagonal - Exemplo



THE
DEVELOPER'S
CONFERENCE



Conhecimento sobre Arquitetura Hexagonal



- Grau de conhecimento sobre DDD:
 - Nenhum
 - Já li / vi palestra a respeito
 - Já fiz uma prova de conceito
 - Já usei profissionalmente



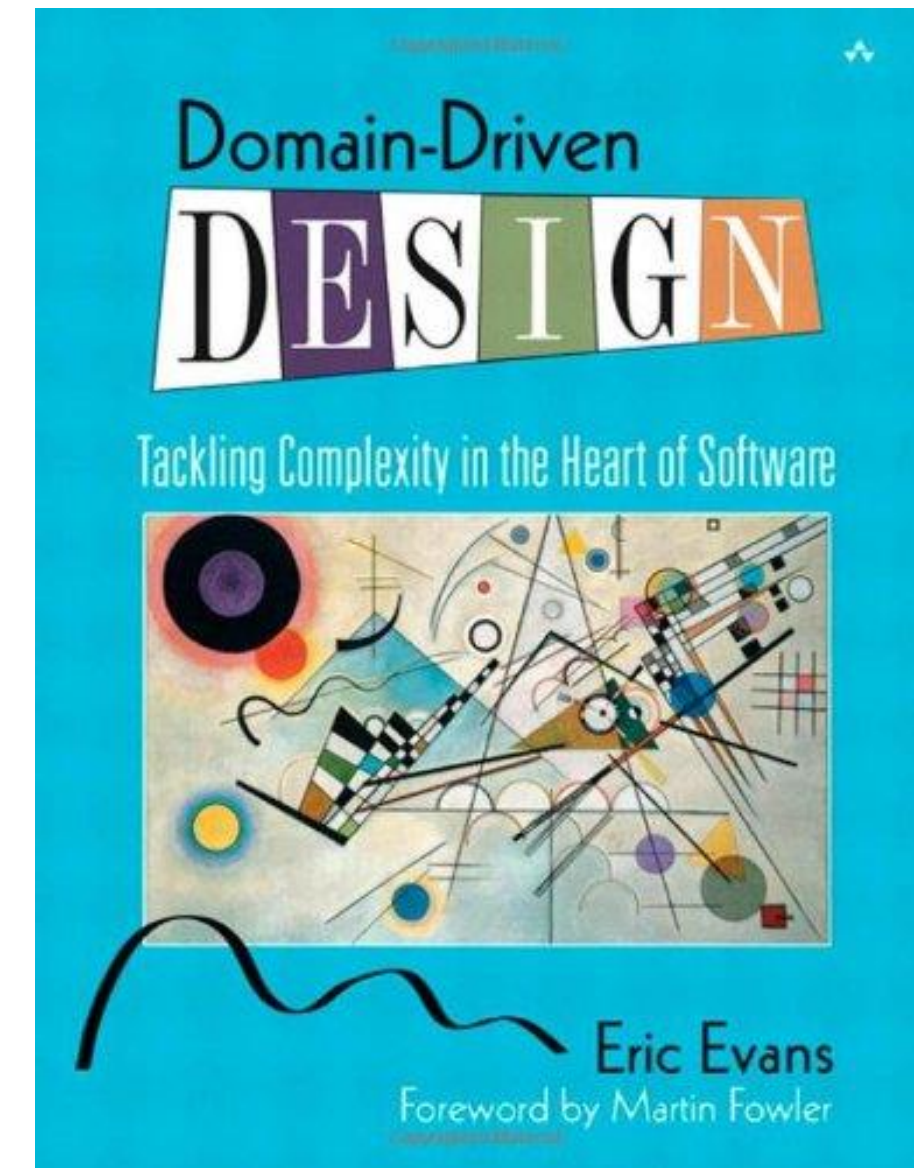
Acesse vevox.app
ID: 156-880-660



THE
DEVELOPER'S
CONFERENCE

Domain Driven Design

- Criada em 2004 por Eric Evans
- “Em sua essência, o domain-driven design é uma maneira de usar modelos para criar software, **especialmente a parte do software que trata regras de negócio complexas em forma de comportamento**”



Domain Driven Design - Conceitos



- Dominio
 - Uma área de conhecimento ou atividade. A area a qual o usuário aplica o programa é o domínio do software
- Modelo
 - Abstrações que descrevem aspectos de um domínio e podem ser usadas para resolver problemas relacionados àquele domínio
- Ubiquitous Language
 - Uma linguagem estruturada ao redor do modelo de domínio e usada pelos membros do time.

Domain Driven Design - Conceitos



- Bounded Context
 - Area que o domínio se aplica. Divisão funcional do sistema
- Entidade
 - Objetos que possuem identidade própria e dados
- Value Object:
 - Objetos que apenas possuem dados, sem identidade própria
- Aggregate Root
 - Entidade que agrega o comportamento geral e mantém a consistência das operações



THE
DEVELOPER'S
CONFERENCE

Domain Driven Design - Exemplo



- Bounded Context:
 - Motor
 - Freio
 - Sistema Direção
- Entidade:
 - Pneu Frontal Direito
- Value Object:
 - Parafuso
- Aggregate Root:
 - Radio

Domain Driven Design - Exemplo



THE
DEVELOPER'S
CONFERENCE

A aplicação deve gerenciar as **contas correntes** dos **usuários**, permitindo **depósitos**, **saques** e **transferências**, registrando todas estas transações para gerar o **extrato** e permitir a **consulta do saldo**.



Entidades

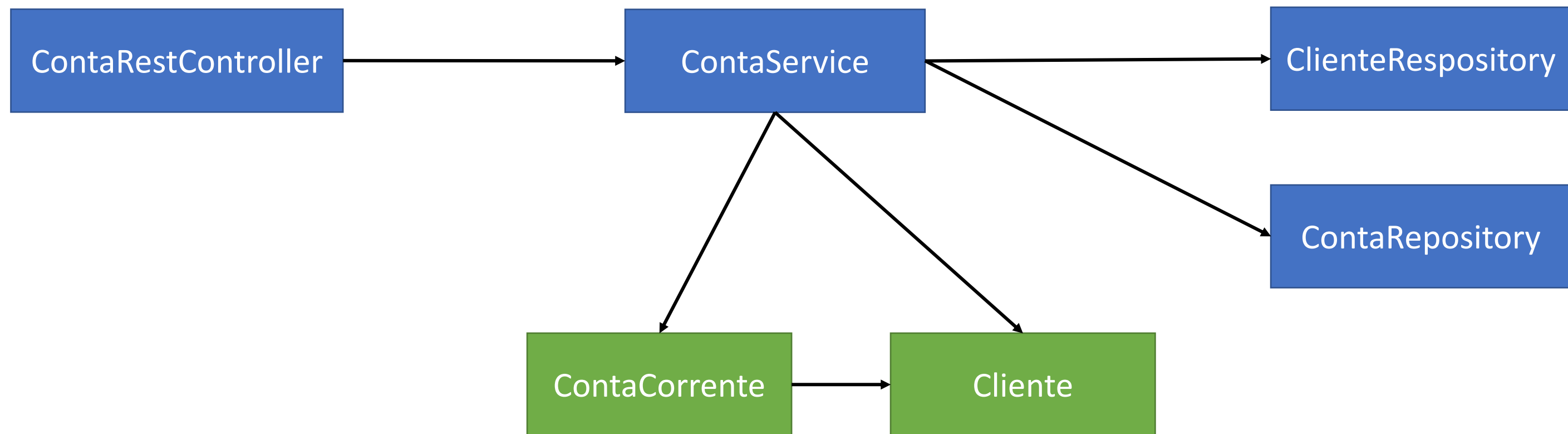


Operações

Domain Driven Design - Exemplo



THE
DEVELOPER'S
CONFERENCE



Domain Driven Design - Pergunta



THE
DEVELOPER'S
CONFERENCE

- Onde devem estar as regras de negócio das operações em um Desenvolvimento DDD?
 - Controller
 - Entidades / Aggregate Root
 - Services
 - Repositórios



Acesse vevox.app
ID: 156-880-660

Dominios Anêmicos

- Entidades (classes de domínio) tem apenas Getters e Setters
- Maior parte da lógica de negócio está nos Services

“O horror fundamental deste anti-pattern é que é tão contrário à noção básica de design orientado a objetos, que é de combinar dados e processos juntos. Modelos de domínio anêmicos são na realidade apenas um design com estilo funcional”

Domain Driven Design – Dominio Ricos



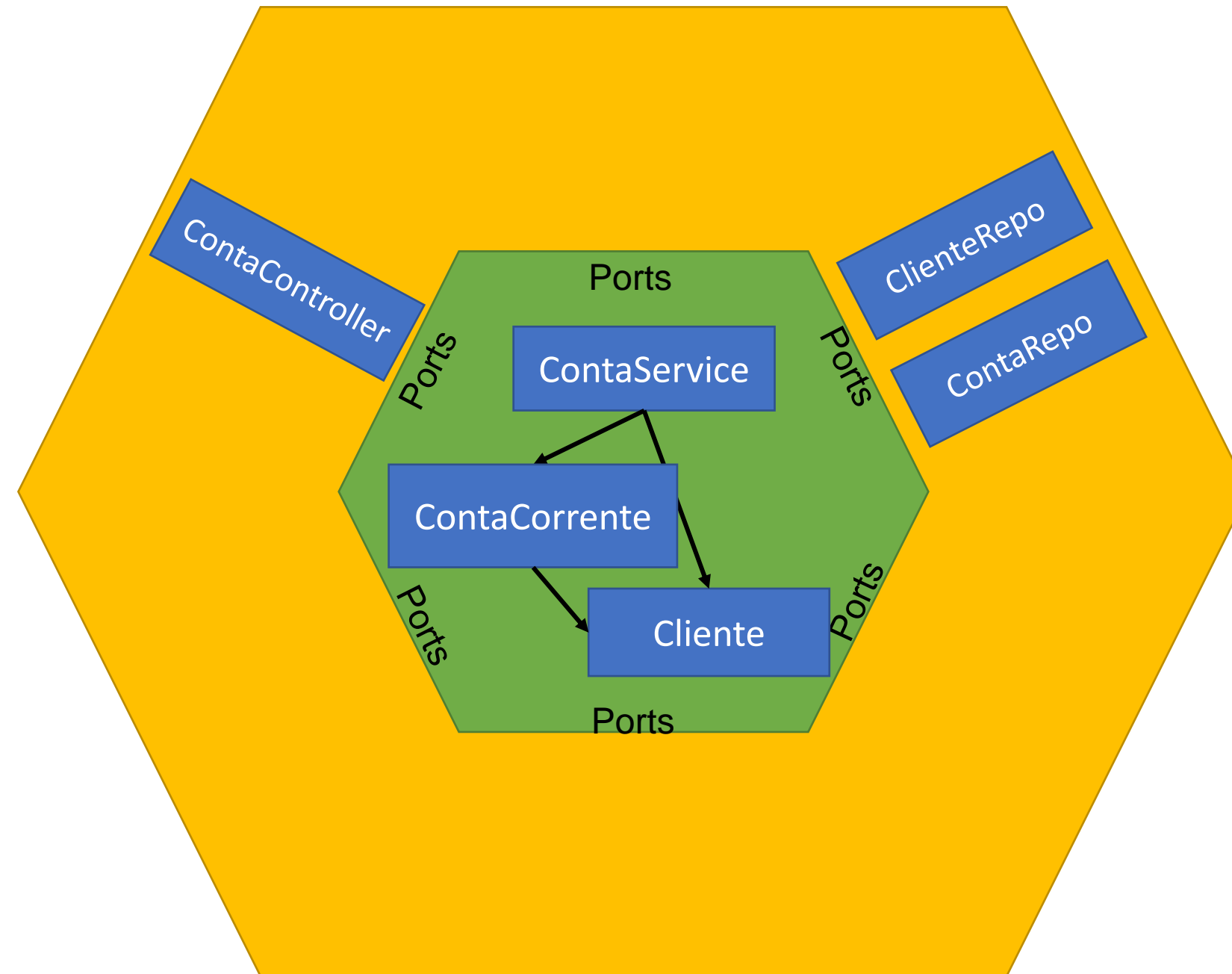
THE
DEVELOPER'S
CONFERENCE

“Camada de aplicação (camada de serviço): Define o trabalho que o software deve fazer e direciona os objetos de domínio expressivos para resolver o problema. A tarefa que esta camada é responsável é significative par ao negócio ou necessária para a interação com as camadas de aplicação de outros sistemas. Esta camada é mantida leve. Ela não contém regras de negócio ou conhecimento, mas apenas coordena tarefas e delega o trabalho para a colaboração dos objetos de domínio na próxima camada.”

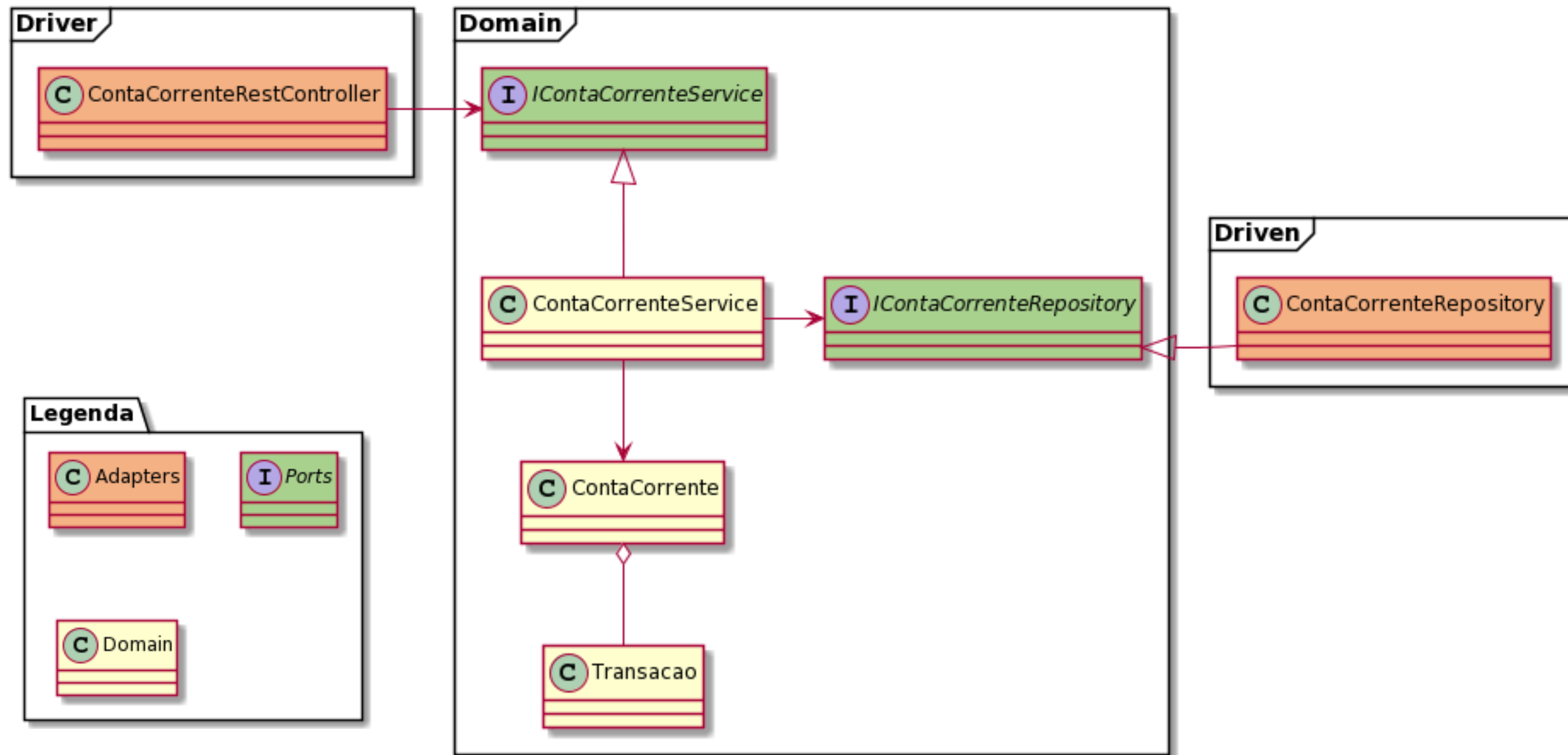
Domain Driven Design + Arquitetura Hexagonal



THE
DEVELOPER'S
CONFERENCE



Como fazer o isolamento?





THE
DEVELOPER'S
CONFERENCE

“Talk is cheap, show me the code!”

Linus Torvalds



THE
DEVELOPER'S
CONFERENCE

Obrigado!

<https://www.linkedin.com/in/aracz>

<https://github.com/andreracz/tdcRecife2020-arquitetura-hexagonal>