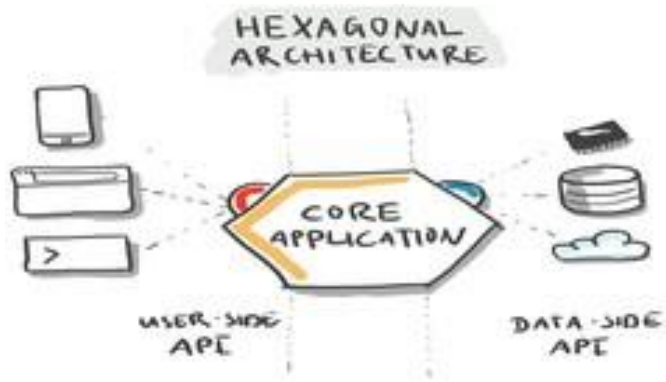




*Crie sua aplicação para funcionar sem ter nem uma UI ou base de dados, ao qual você pode rodar testes de regressão automatizado, implemente quando a base de dados não está disponível e conecte aplicações sem envolver elas.*

*(Alistair Cockburn)*



# ARQUITETURA HEXAGONAL: UMA INTRODUÇÃO

# Morvana Bonin

- ▷ Desenvolvedora backend na Sensedia
- ▷ Analista de sistema formada pela faculdade SENAC-RS de Porto Alegre
- ▷ Curiosa e apaixonada por aprender desde tecnologia até a entender certos comportamentos humanos.
- ▷ Otaku

contatos:

@morvanabonin

gmail.com/twitter/insta/gitlab



# Microservices

## Porto Alegre



Não existe bala  
de prata!



*Uma pessoa imatura pensa que todas suas  
escolhas geram ganhos...*

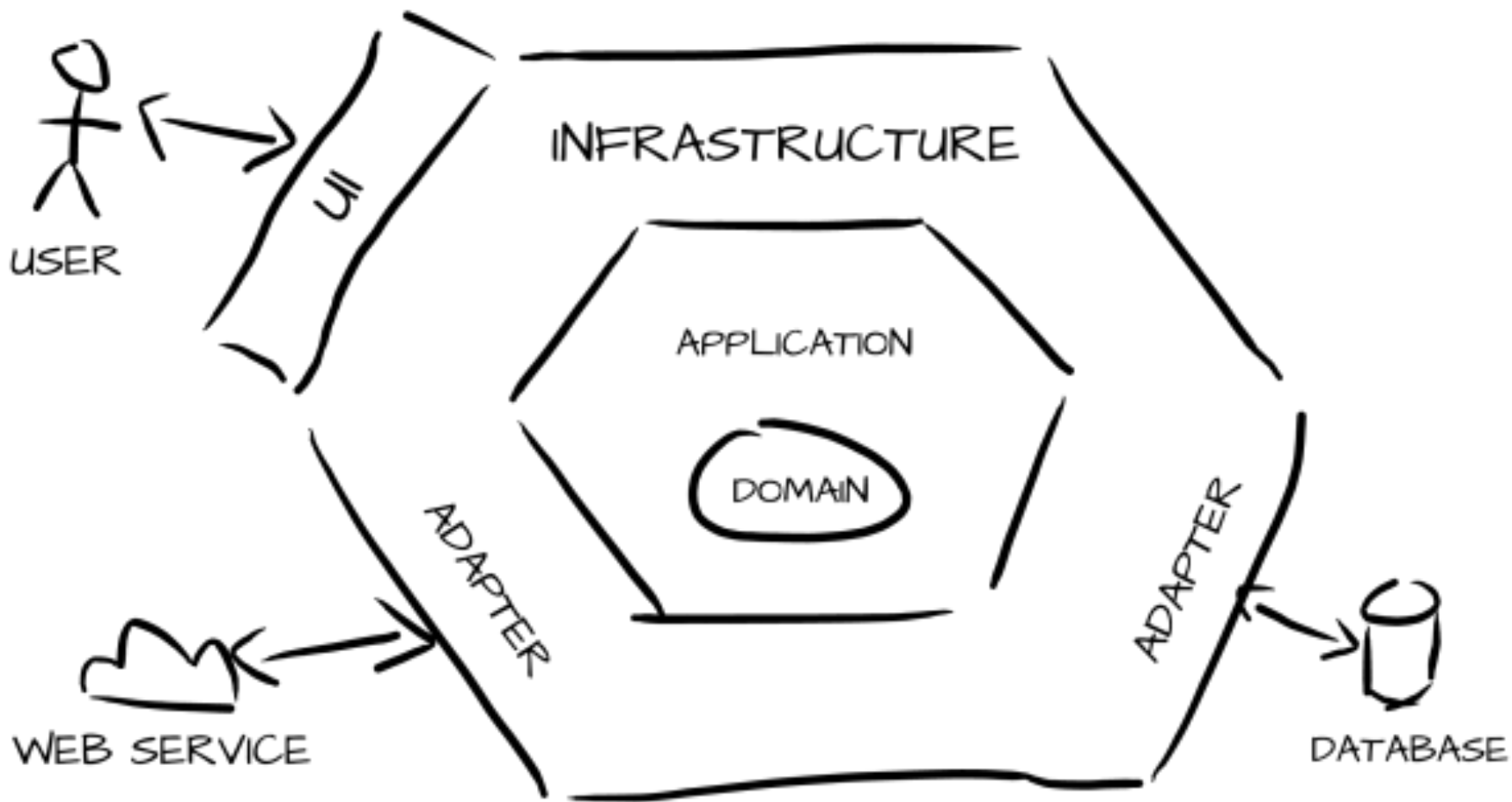
*uma pessoa madura sabe que todas as  
escolhas tem perdas.*

*(Augusto Cury)*

1.

# Arquitetura Hexagonal

Afinal o que é?

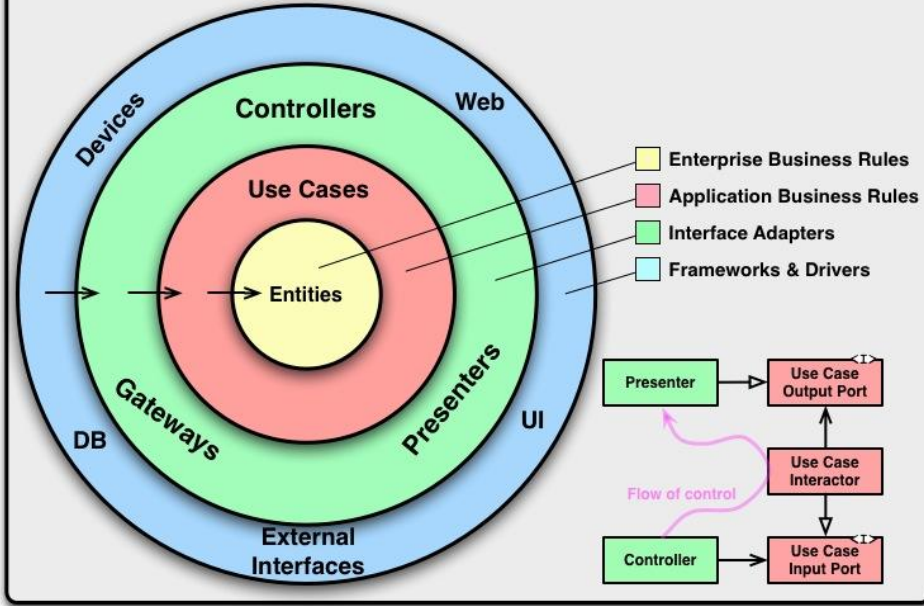




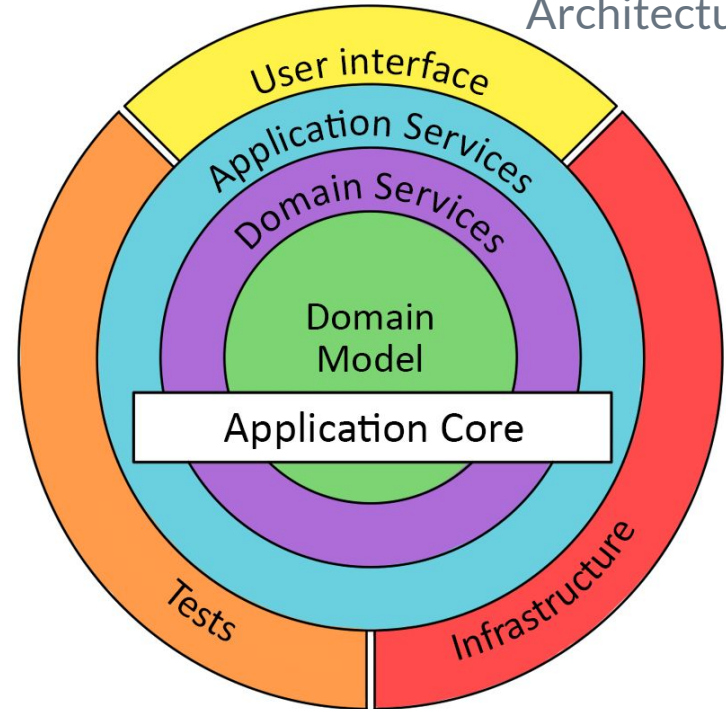


- ▷ Criada em 2005 por Alistair Cockburn
- ▷ Também conhecida como padrão de portas e adaptadores
- ▷ É uma guideline que serve de referência para construção de de software.

# The Clean Architecture



# Onion Architecture



# Motivação

- ▶ Dificuldade na construção e implementação dos testes
- ▶ Dificuldade na manutenção de código.
- ▶ Evolução da aplicação devido a alta dependência com fatores externos.

## Natureza da Solução

Separar a parte interna da externa, ou seja, regras de negócio na base de dados ou na interface do usuário.



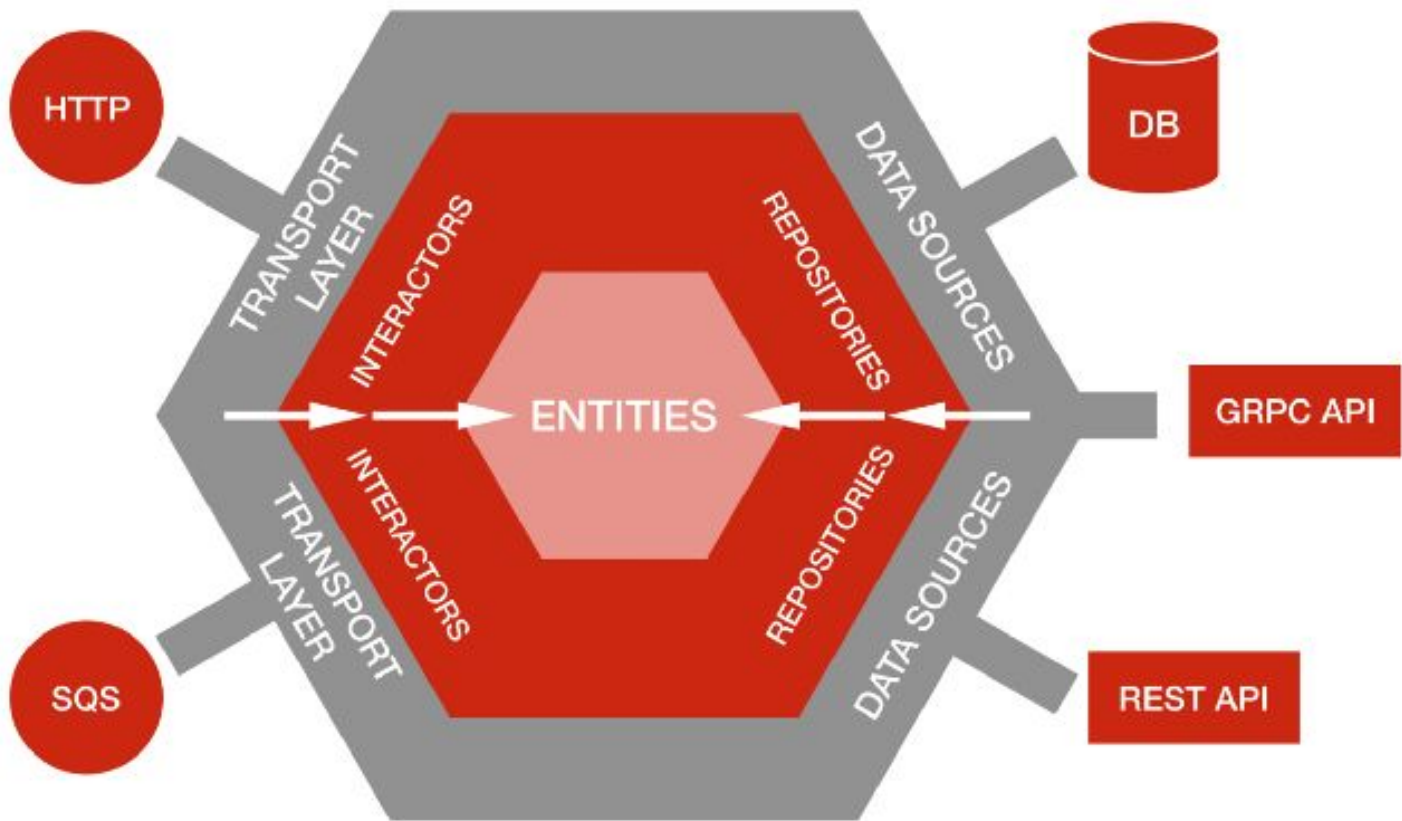


## Separation of Concerns (SoC)

Separar conceitos de forma modular e com cada módulo se atendo a resolver apenas um único problema.

A arquitetura hexagonal tem o seguinte princípio de dependências:

**Somente de fora para dentro!**



2.

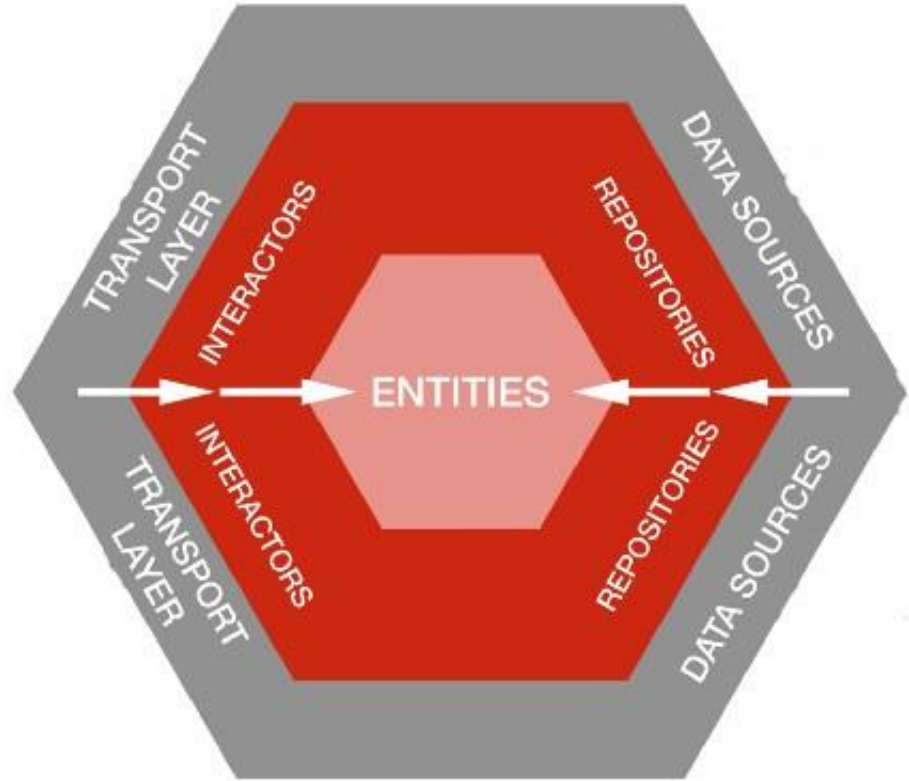
# Centro do Hexágono

**Domínio**



Foco no negócio

Seu principal objetivo é permitir que um software seja criado e testado focando na regra de negócios e ignorando dependências externas



# No centro do Hexágono temos

- ▷ Domínios/ Entidades
- ▷ Interfaces (farão o papel de portas)
- ▷ Services (Classes)
- ▷ Exceptions do sistema
- ▷ IoC, Logs

O centro do hexágono não deve ser influenciável por tecnologias externas, bem como interfaces gráficas, interfaces comunicações e dispositivos externos do mundo real.

## Fora do Hexágono

Na parte externa do hexágono nós temos os atores, que serão qualquer interação com o mundo real que nossa aplicação irá interagir.

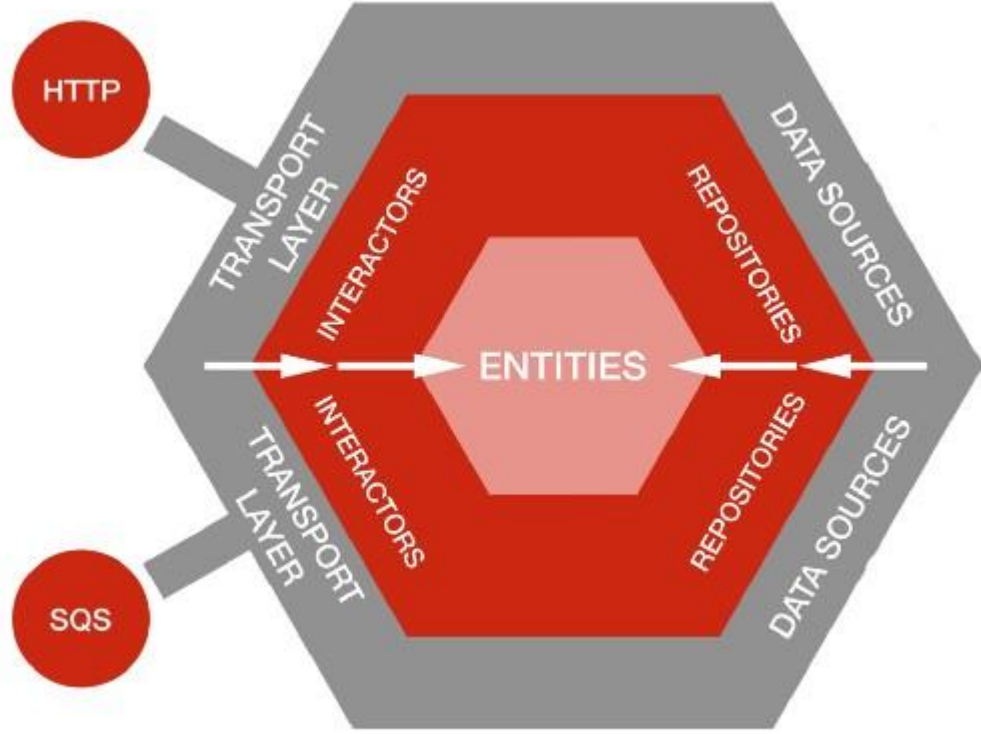
São divididos em 2 grupos:

1. Ator Primário Condutor (Driver)
2. Ator Secundário Conduzido (Driven)

3.

# Lado Esquerdo do Hexágono

Atores primários (Drivers)

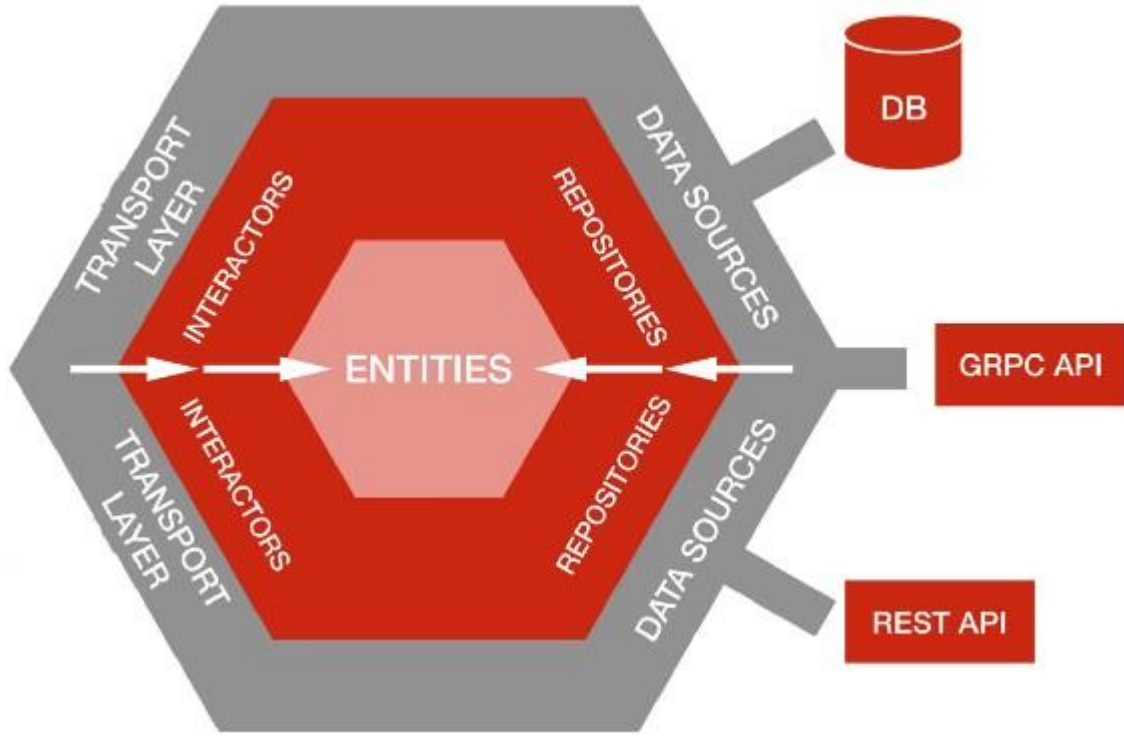


O lado esquerdo compreende o lado permutável e fluído ao qual um ator externo irá **interagir** com a solução, o software. (realiza a ação, **dirige**).

4.

# Lado Direito do Hexágono

Atores secundários (Driven)



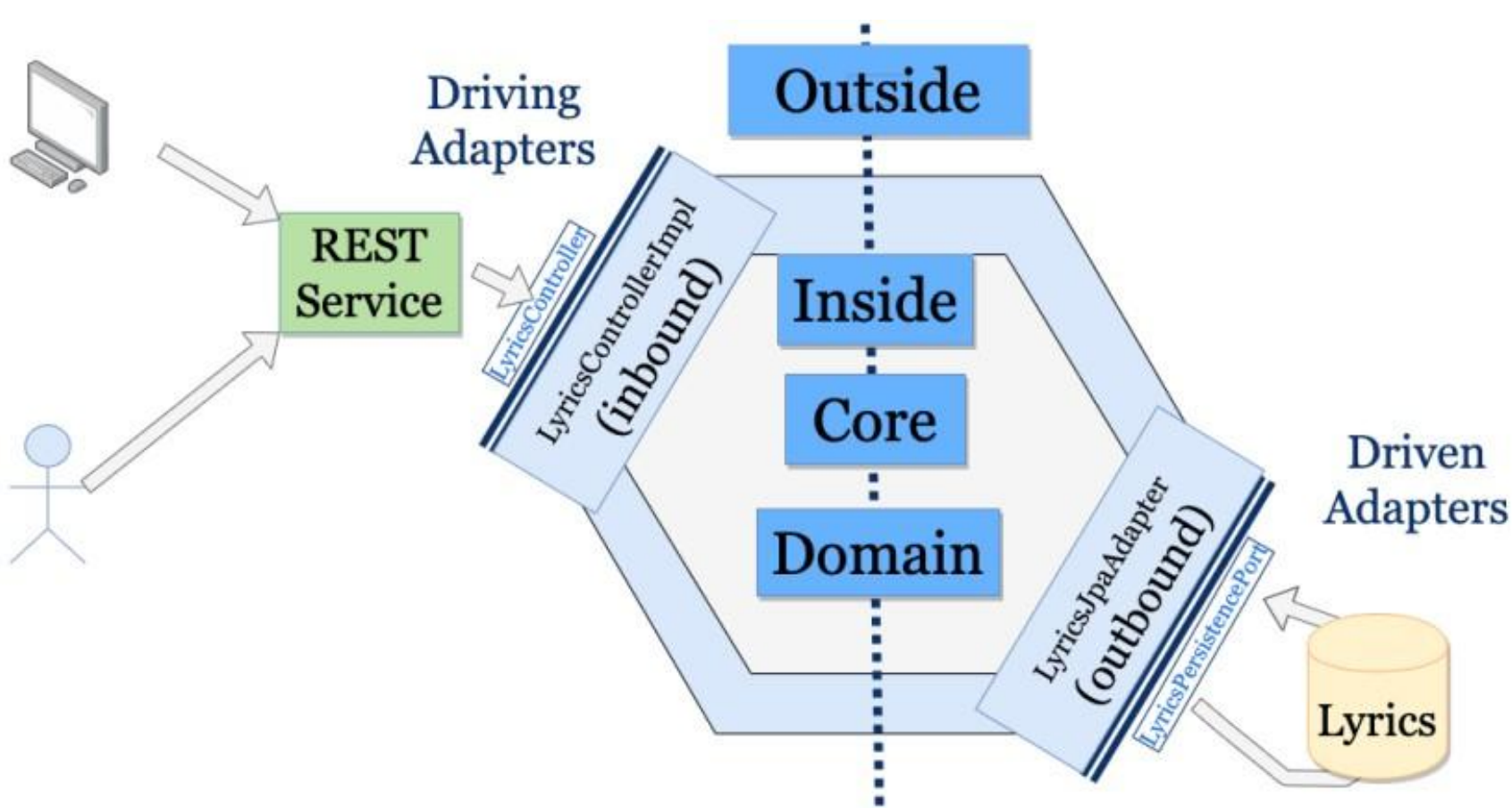


O lado direito compreende o lado permutável e fluído que fornece os serviços de infraestrutura e interage com o banco de dados, sistema de arquivos, etc (**executa** uma ação, **dirigido**)

# Portas

As portas como são chamadas, tem exatamente essa função, ser responsável pela comunicação de fora para dentro dos lado esquerdo e direito para dentro do centro do hexágono.

Na prática, as portas, são frequentemente implementadas como interfaces a serem usadas por terceiros.

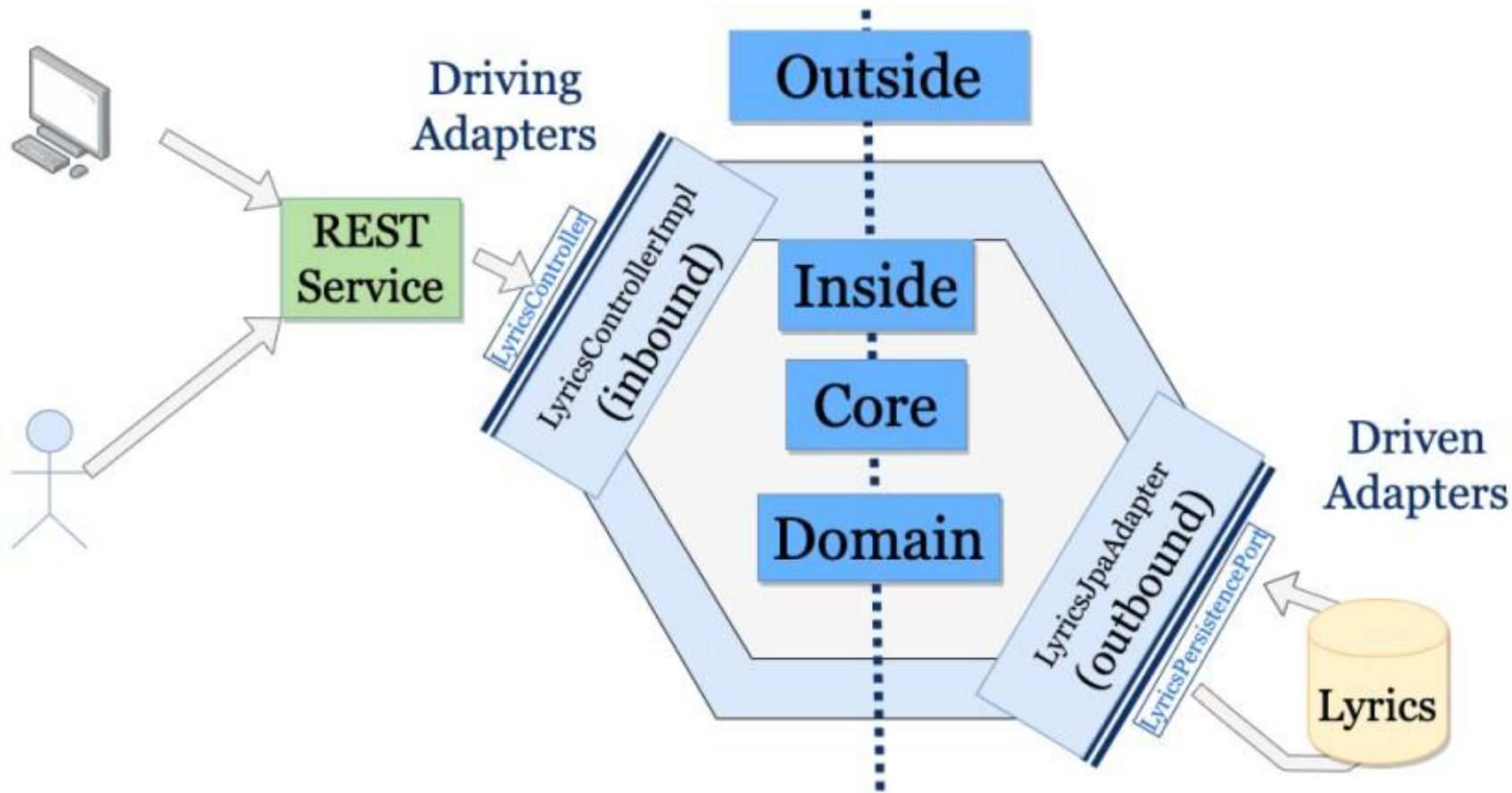


# Adaptadores

Os adaptadores são componentes de softwares para que uma tecnologia externa possa interagir com uma porta do centro hexágono.

Para cada tecnologia que for utilizar, deve ser criado de adaptador que irá interagir com a porta do hexágono.

Na prática, os adaptadores são essencialmente a implementação de portas.



# IoC (Inversion of Control)

## Controle de inversão



é um Pattern que prega para usarmos o controle das instâncias de uma determinada classe ser tratada externamente e não dentro da classe em questão, ou seja, Inverter o controle de uma classe delegando para uma outra classe, interface, componente, serviço, etc.



# Pontos Positivos 😊

- ▶ ~~Solução independente de frameworks~~
- ▶ Solução Independente de Serviços Externos
- ▶ Possível adiar algumas decisões técnicas
- ▶ Criação e substituição de adaptadores
- ▶ Facilidade em testar a aplicação.
- ▶ Facilidade na troca de tecnologias.



# Pontos Negativos 🙄

- ▷ Complexidade Extra
- ▷ Custo de criação e manutenção.
- ▷ Não existe uma orientação sobre organização de código (diretórios, camadas)



# Dicas

Projetos que fazem uso de regras de negócio em banco de dados ou “amarrados” a tecnologias de terceiros ou proprietários não justifica o uso dessa abordagem.

Projetos de **médio e grande porte** que tem de longo ciclo de vida e precisam ser modificados muitas vezes, principalmente por crescimento e escalabilidade, além de não serem amarrados com o *frontend* e nem *back services* justificam o uso dessa abordagem.

Arigatō gozaimasu!

ありがとうございます

**Perguntas?**

# Referências e Links Úteis

Arquitetura Hexagonal com Java

<https://www.udemy.com/course/arquitetura-hexagonal-com-java-1>

Hexagonal architecture

<https://alistair.cockburn.us/hexagonal-architecture>

Alistair in the "Hexagone"

<https://www.youtube.com/playlist?list=PLGl1Jc8ErU1w27y8-7Gdcloy1tHO7NriL>

Arquitetura Hexagonal: O que você precisa saber

<https://youtu.be/or5zAOASPjU>

Arquitetura Hexagonal

<https://fernandofranzini.wordpress.com/2019/04/09/arquitetura-hexagonal>

DDD, Hexagonal, Onion, Clean, CQRS, ... How I put it all together

<https://herbertograca.com/2017/11/16/explicit-architecture-01-ddd-hexagonal-onion-clean-cqrs-how-i-put-it-all-together>

# Referências e Links Úteis - Cont.

Entendendo a Separação de Conceitos

<https://www.andrecelestino.com/entendendo-a-separacao-de-conceitos>

Hexagonal Architecture

<https://stefanoalletti.wordpress.com/2017/10/27/hexagonal-architecture>

Ready for changes with Hexagonal Architecture

<https://netflixtechblog.com/ready-for-changes-with-hexagonal-architecture-b315ec967749>

Hexagonal Architecture Gitlab Code

<https://gitlab.com/hexagonal-architecture-java>

Hexagonal Architecture / Ports and Adapters

<https://dev.to/jofisaes/hexagonal-architecture-ports-and-adapters-1h4m>