Clean Architecture!

A forma de escalar a
manutenção do código

alexrios.dev


BLACK
LIVES
MATTER

Whoa!

Como que isso se dá no meu código?

Onde posso usar?

01

Back

Servidores

02

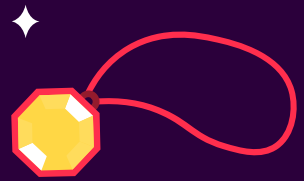
Mobile

Android, IOS

03

Front

Navegador



“Clean Architecture também é conhecida por arquitetura por camadas”

—Dev malandro

01 Exemplo

Porque só teorizar não rola

Uma loja online

Imagina que uma loja online utiliza um banco de dados postgres para persistir o seus dados.

Agora eu preciso fazer full text search

Postgres

Mantemos os dados estruturados nele.

ElasticSearch

Fazer o ranking de produtos de forma simples pois é a solução adequada.

QUal o problema quando eu tenho esse tipo de mudança com MVC?

Model

View

Controller

Não é clean

O modelo sabe da infraestrutura.

Logo temos mais de uma preocupação aqui.

Logica de dominio + persistência.

Cai na real! Quem que fica trocando base de dados?

Bom ponto! Mas as vantagens são maiores que apenas esse tipo de alteração.

Prepare-se para o futuro

Libs

É mais simples a atualização de libs.

Frameworks

É mais simples trocar por outros frameworks

Infraestrutura

Alterar o ambiente onde sua App funciona.

Quais arquiteturas limpas mais famosas?

DCI

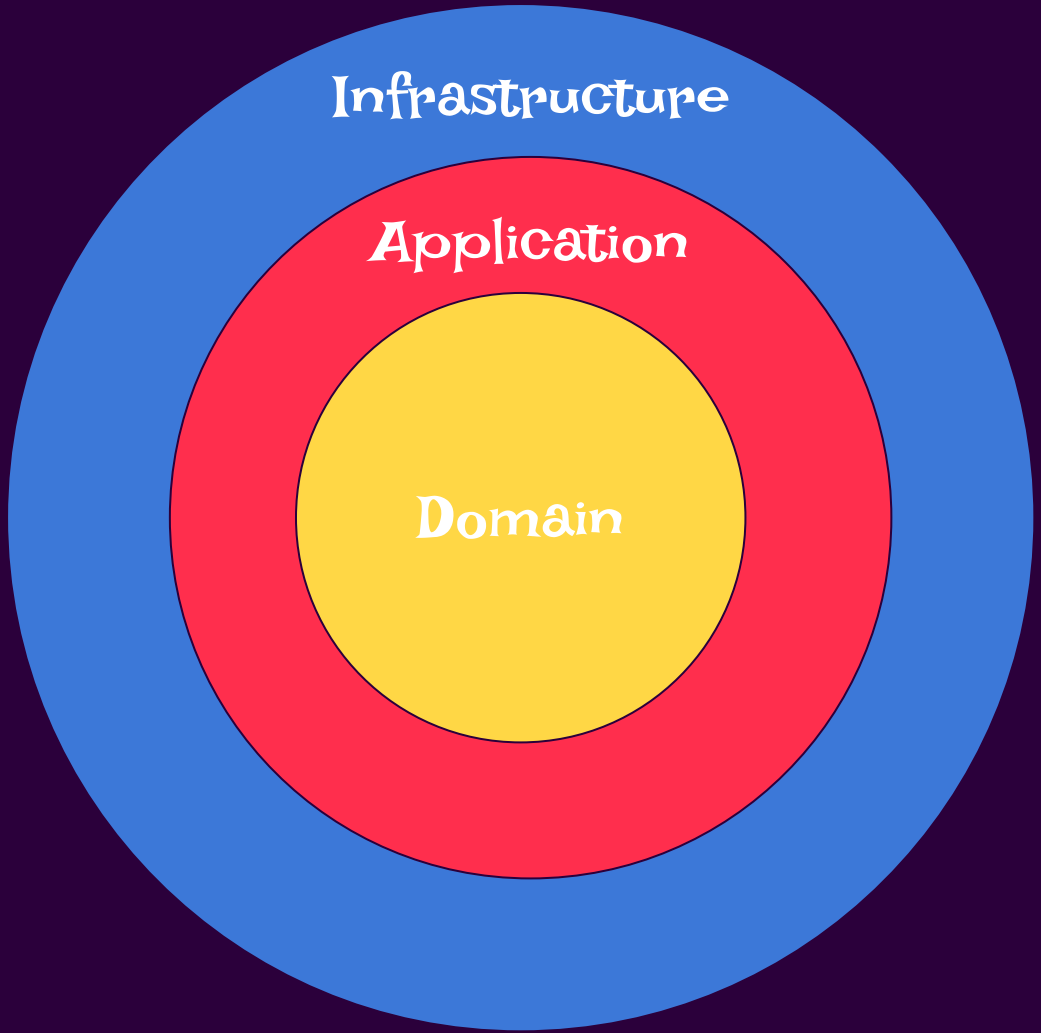
Onion

Hexagonal

Screaming

Tudo gira em torno dos mesmos conceitos

Camadas


Infrastructure
Application
Domain

Regra de dependência

O camada de fora só pode depender das camadas de dentro

A camada de dentro NUNCA pode depender de uma camada de fora

Application

Os casos de uso da nossa aplicação.

Ex: ProductSearcher

Domain

As Entidades da nossa aplicação.
(do que se trata o negócio)

Ex: Product

Infrastructure

Tudo que lida com I/O nossa aplicação.
(conexões, http, filas, bancos de dados)

Ex: ProductRepository

02

Implementação

Aqui que o bicho pega.

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

User

- UserID
- Name
- LastName

UserRegistrar

- Rename

Product

- ProductID
- Name
- Description

ProductCreator

- Create

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

UserRegistrar

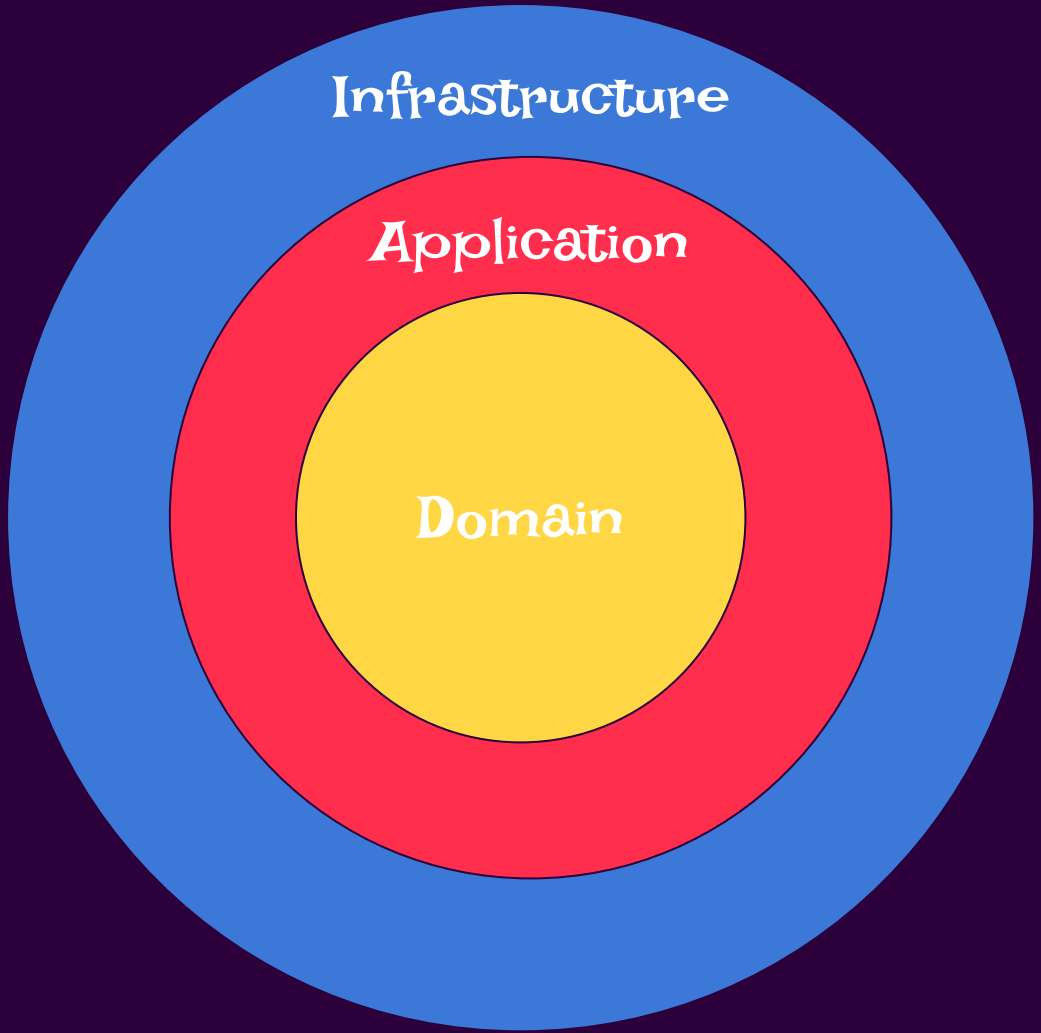
- Rename

UserRepository

ProductCreator

- Create

ProductRepository


Infrastructure
Application
Domain

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

UserRegistrar
- Rename

UserRepository

ProductCreator
- Create

ProductRepository

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

UserRegistrar

- Rename

UserRepository

ProductCreator

- Create

ProductRepository

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

UserRegistrar

- Rename

UserRepository

ProductCreator

- Create

ProductRepository

User

- UserID
- Name
- LastName

UserRegistrar

- Rename

UserRepository

Product

- ProductID
- Name
- Description

ProductCreator

- Create

ProductRepository

User

- UserID
- Name
- LastName

UserRegistrar

- Rename

UserRepository

Product

- ProductID
- Name
- Description

ProductCreator

- Create

ProductRepository

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

UserRegistrar

- Rename

UserRepository

ProductCreator

- Create

ProductRepository

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

UserRegistrar

- Rename

UserRepository

ProductCreator

- Create

ProductRepository

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

UserRegistrar
- Rename

UserRepository

ProductCreator
- Create

ProductRepository

ProductNotFound

ProductCreated

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

ProductNotFound

ProductCreated

UserRegistrar
- Rename

UserRepository

ProductCreator
- Create

ProductRepository

UserController

PostgresUserRepository

ProductController

ElasticProductRepository

User

- UserID
- Name
- LastName

Product

- ProductID
- Name
- Description

UserRegistrar
- Rename

UserRepository

ProductCreator
- Create

ProductRepository

ProductNotFound

ProductCreated

UserController

PostgresUserRepository

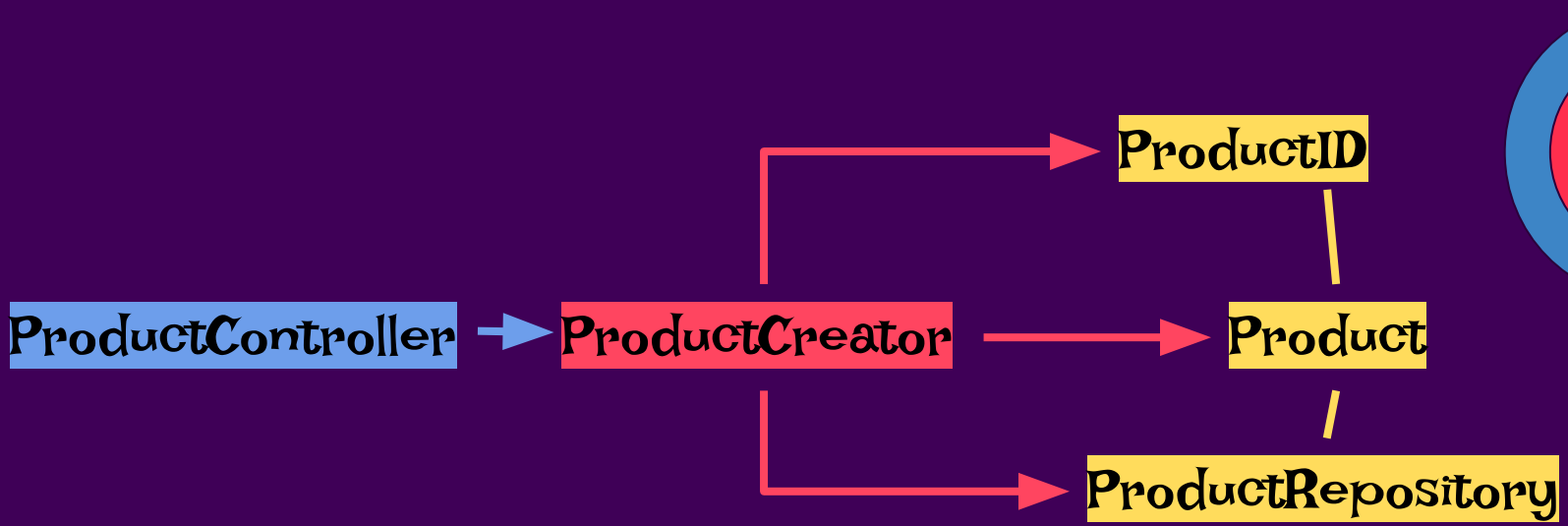
ProductController

ElasticProductRepository

ProductController → ProductCreator → Product

ProductCreator → ProductID

ProductCreator → ProductRepository




```
interface ProductRepository {
    void save(Product product)
}
```

ProductID

ProductController → ProductCreator → Product

ProductCreator → ProductRepository

ElasticProductRepository → ProductRepository

ProductID

Product

ProductController

ProductCreator

ProductRepository

ElasticProductRepository

UserID

User

UserController

UserRegistrar

UserRepository

PostgresUserRepository

Hora de arrumar os pacotes

Quem é?

Conceito

Que tipo?

Product/

ProductID

Product

ProductController

ProductCreator

ProductRepository

ElasticProductRepository

User/

UserID

User

UserController

UserRegistrar

UserRepository

PostgresUserRepository

Por conceito

[Pros]
Facil de navegar

[Contras]
Bagunça dentro do package
Explosão de arquivos

Controller/

ProductController

ElasticProductRepository

UseCase/

ProductCreator

UserRegistrar

ValueObject/

ProductID

UserID

Entity/

User

Product

Repositories/

UserController

PostgresUserRepository

Exception/

InvalidUserEmail

ProductNotExist

Repository/

ProductRepository

UserRepository

Quem é

[Pros]

💀💀💀

[Contras]

Não diz nada

Infrasctructure/

ProductController

ElasticProductRepository

PostgresUserRepository

UserController

Application/

ProductCreator

UserRegistrar

Domain/

ProductID

Product

ProductRepository

UserID

User

UserRepository

Que tipo

[Pros]
Atende perfeitamente a projetos pequenos.

[Contras]
Não escala bem em projetos grandes pela explosão de arquivos

Infrasctructure/

Product/

ProductController

ElasticProductRepository

User/

PostgresUserRepository

UserController

Application/

Product/

ProductCreator

User/

UserRegistrar

Domain/

ProductID

Product

ProductRepository

UserID

User

UserRepository

Que tipo + Quem é

[Pros]
Funciona na arquitetura hexagonal e com DDD

[Contras]
Falta coesão nos pacotes

User/

Application/

UserRegistrar

Domain/

UserID

User

UserRepository

Infrastructure/

UserController

PostgresUserRepository

Quem é + Que tipo

Product/

Application/

ProductCreator

Domain/

ProductID

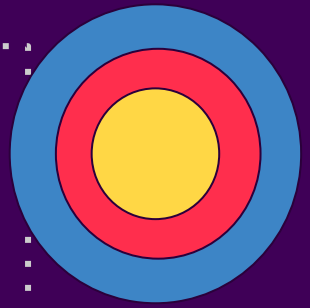
Product

ProductRepository

Infrastructure/

ProductController

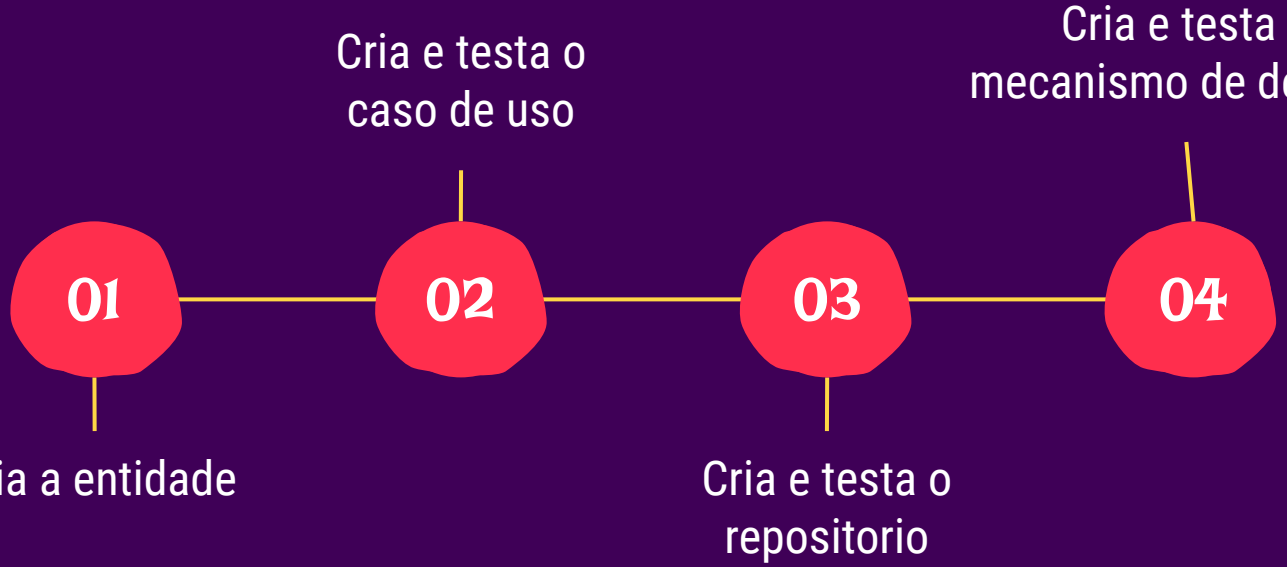
ElasticProductRepository



[Pros]
Mais semântica, escalabilidade e manutenção.

[Contras]
Causa brain damage em quem usava MVC

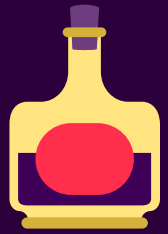
Construindo com Clean Architecture



Thanks

alexrios.dev

Perguntas?



CREDITS: This presentation template was created by **Slidesgo**,
including icons by **Flaticon** and infographics & images by **Freepik**
Please keep this slide for attribution

Os exemplos foram retirados dos materiais do @rafaoe da codely.tv