



# Implementando Clean Arch

**Um pouco de prática!**



# Implementando Clean Arch

Em parceria com o minicurso de Testes Automatizados:

## Um pouco de prática!

**Testes automatizados de software: vantagens, desafios e técnicas de escrita - Minicurso**

**Data:** 21/10/2022  
**Início:** 08:00  
**Fim:** 12:00  
**Local:** Laboratório central 6B

Este minicurso irá apresentar os principais conceitos e práticas sobre a escrita de testes automatizados de software, abordando suas vantagens e desafios.

Pré-requisitos: é desejável que a pessoa interessada tenha conhecimentos iniciais em: programação orientado a objetos, alguma linguagem de programação, as atividades práticas serão passadas em C# porém os conceitos podem ser transferidos para qualquer linguagem da preferência da pessoa interessada.

Materiais necessários: de preferência Visual Studio 2019 com a versão 5.0 do .net. Caso a pessoa queira realizar a atividade prática em outra linguagem a pessoa será responsável por ter uma IDE ou editor de texto de sua preferência já instalada.

**Thiago Aires Geraldo de Almeida**

✉ [thiago.almeida@lambda3.com.br](mailto:thiago.almeida@lambda3.com.br)

# Implementando Clean Arch

Em parceria com o minicurso de Testes Automatizados:

## Um pouco de prática!

Desenvolver um carrinho de compras de uma loja virtual onde é possível adicionar N produtos com N quantidades, ao finalizar a compra o carrinho deve passar o valor total da compra da seguinte forma:

Produto com valor unitário acima de 100 reais deve aplicar um desconto de 10% no valor unitário;

Produto com mais de 10 quantidades (10 produtos iguais) deve aplicar um desconto de 10% no valor total do mesmo produto;

Compras com valor total acima de 5000 reais deve aplicar um desconto de 20% do valor;

Todos os descontos podem ser acumulados.

**Testes automatizados de software: vantagens, desafios e técnicas de escrita - Minicurso**

**Data:** 21/10/2022  
**Início:** 08:00  
**Fim:** 12:00  
**Local:** Laboratório central 6B

Este minicurso irá apresentar os principais conceitos e práticas sobre a escrita de testes automatizados de software, abordando suas vantagens e desafios.

Pré-requisitos: é desejável que a pessoa interessada tenha conhecimentos iniciais em: programação orientado a objetos, alguma linguagem de programação, as atividades práticas serão passadas em C# porém os conceitos podem ser transferidos para qualquer linguagem da preferência da pessoa interessada.

Materiais necessários: de preferência Visual Studio 2019 com a versão 5.0 do .net. Caso a pessoa queira realizar a atividade prática em outra linguagem a pessoa será responsável por ter uma IDE ou editor de texto de sua preferência já instalada.

**Thiago Aires Geraldo de Almeida**

✉ [thiago.almeida@lambda3.com.br](mailto:thiago.almeida@lambda3.com.br)



# Implementando Clean Arch

## Um pouco de prática!

Desenvolver um carrinho de compras de uma loja virtual onde é possível adicionar N produtos com N quantidades, ao finalizar a compra o carrinho deve passar o valor total da compra da seguinte forma:

Produto com valor unitário acima de 100 reais deve aplicar um desconto de 10% no valor unitário;

Produto com mais de 10 quantidades (10 produtos iguais) deve aplicar um desconto de 10% no valor total do mesmo produto;

Compras com valor total acima de 5000 reais deve aplicar um desconto de 20% do valor;

Todos os descontos podem ser acumulados.



# Implementando Clean Arch

## Um pouco de prática!

Desenvolver um carrinho de compras de uma loja virtual onde é possível adicionar N produtos com N quantidades, ao finalizar a compra o carrinho deve passar o valor total da compra da seguinte forma:

Produto com valor unitário acima de 100 reais deve aplicar um desconto de 10% no valor unitário;

Produto com mais de 10 quantidades (10 produtos iguais) deve aplicar um desconto de 10% no valor total do mesmo produto;

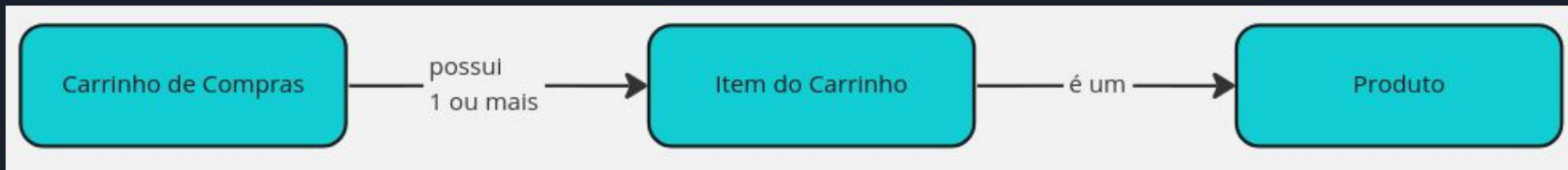
Compras com valor total acima de 5000 reais deve aplicar um desconto de 20% do valor;

Todos os descontos podem ser acumulados.

# Implementando Clean Arch

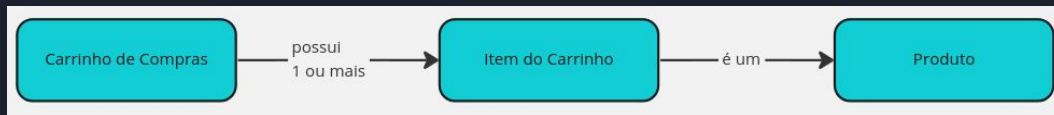
## Um pouco de prática!

Desenvolver um carrinho de compras de uma loja virtual onde é possível adicionar N produtos com N quantidades, ao finalizar a compra o carrinho deve passar o valor total da compra da seguinte forma:



# Implementando Clean Arch

## Um pouco de prática!



Desenvolver um carrinho de compras de uma loja virtual onde é possível adicionar N produtos com N quantidades, ao finalizar a compra o carrinho deve passar o valor total da compra da seguinte forma:

**Produto com valor unitário acima de 100 reais deve aplicar um desconto de 10% no valor unitário;**

**Produto com mais de 10 quantidades (10 produtos iguais) deve aplicar um desconto de 10% no valor total do mesmo produto;**

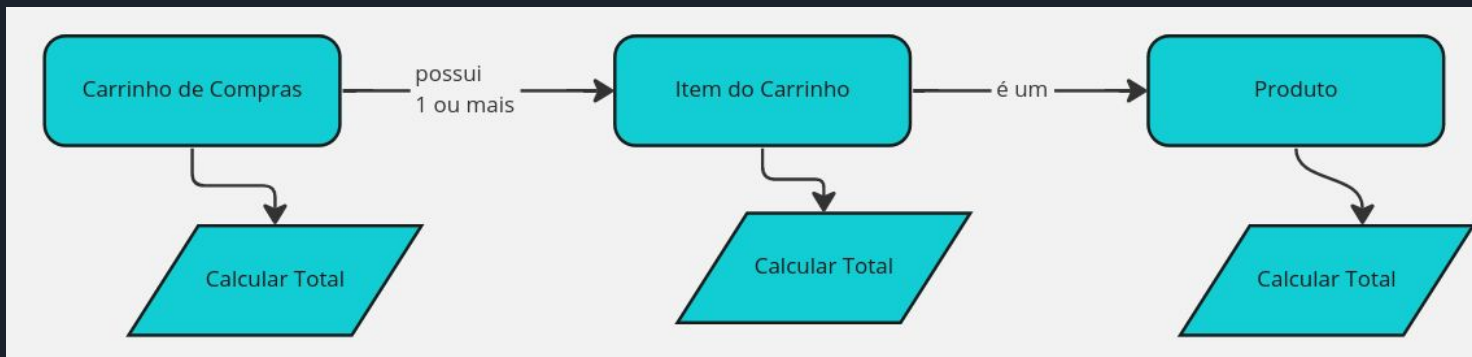
**Compras com valor total acima de 5000 reais deve aplicar um desconto de 20% do valor;**

**Todos os descontos podem ser acumulados.**

# Implementando Clean Arch

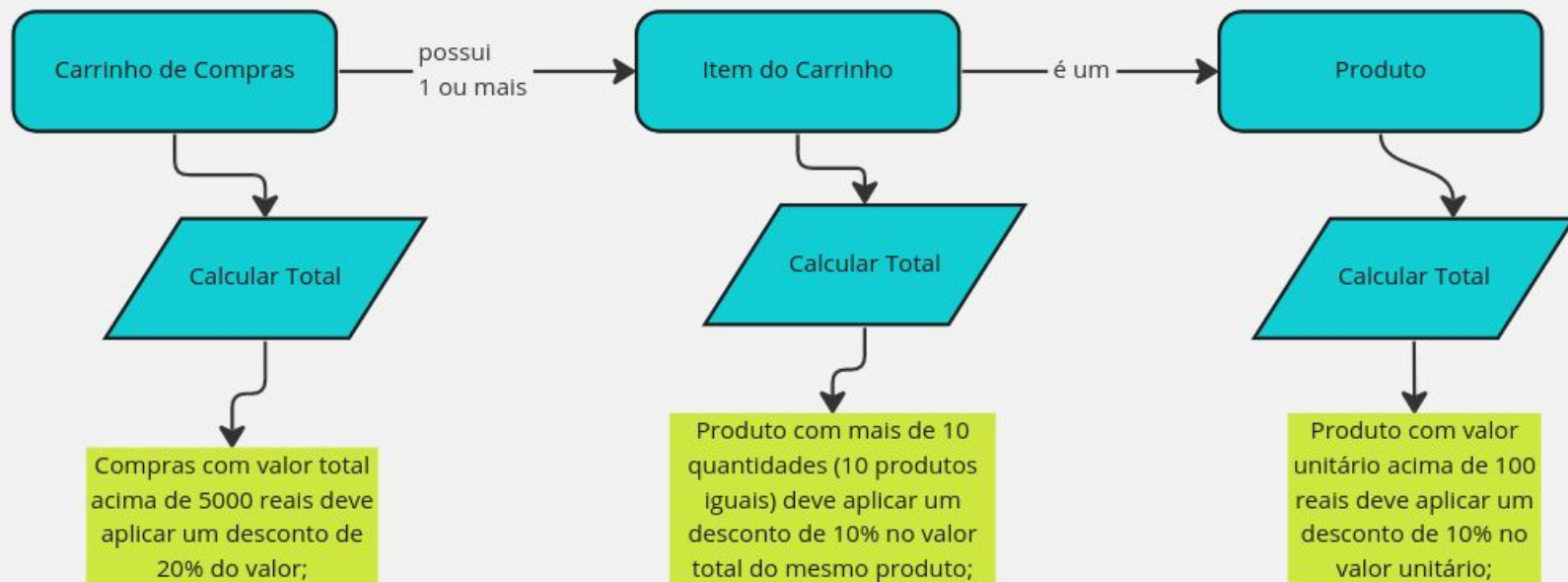
## Um pouco de prática!

Desenvolver um carrinho de compras de uma loja virtual onde é possível adicionar N produtos com N quantidades, ao finalizar a compra o carrinho deve passar o valor total da compra da seguinte forma: Produto com valor unitário acima de 100 reais deve aplicar um desconto de 10% no valor unitário; Produto com mais de 10 quantidades (10 produtos iguais) deve aplicar um desconto de 10% no valor total do mesmo produto; Compras com valor total acima de 5000 reais deve aplicar um desconto de 20% do valor; Todos os descontos podem ser acumulados.





# Implementando Clean Arch



Todos os descontos podem ser acumulados.

# Implementando Clean Arch

Desenvolver um carrinho de compras de uma loja virtual onde é possível adicionar N produtos com N quantidades, ao finalizar a compra o carrinho deve passar o valor total da compra da seguinte forma:

**...onde é possível adicionar N produtos com N quantidades...**

**Adicionar Item no Carrinho**

**Input:**  
**Output:**

**...ao finalizar a compra o carrinho deve passar o valor total...**

**Checkout**

**Input:**  
**Output:**

# Implementando Clean Arch

Desenvolver um carrinho de compras de uma loja virtual onde é possível adicionar N produtos com N quantidades, ao finalizar a compra o carrinho deve passar o valor total da compra da seguinte forma:

...onde é possível adicionar N produtos com N quantidades...

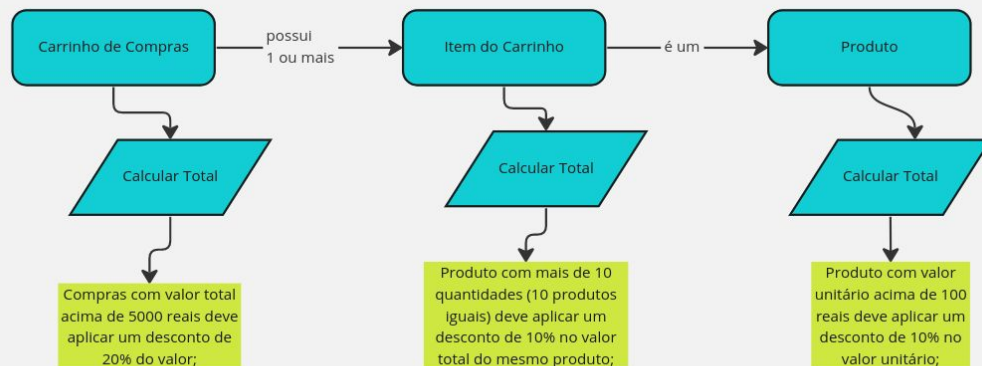
**Adicionar Item no Carrinho**

**Input:**  
**Output:**

...ao finalizar a compra o carrinho deve passar o valor total...

**Checkout**

**Input:**  
**Output:**



Todos os descontos podem ser acumulados.

# Implementando Clean Arch

...onde é possível adicionar N produtos com N quantidades...

## Adicionar Item no Carrinho

**Input:** Produto, Quantidade.

**Output:** Valor total do item.

### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

...ao finalizar a compra o carrinho deve passar o valor total...

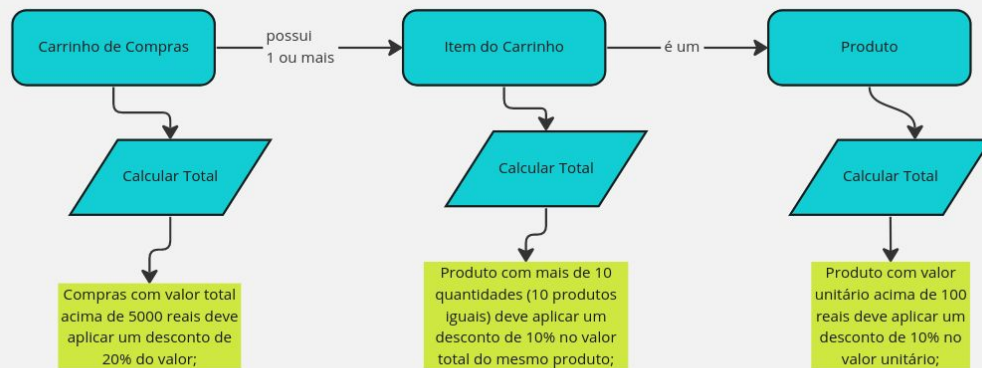
## Checkout

**Input:** Carrinho

**Output:** Valor total do carrinho

### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total



Todos os descontos podem ser acumulados.

# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?

...onde é possível adicionar N produtos com N quantidades...

## Adicionar Item no Carrinho

**Input:** Produto, Quantidade.

**Output:** Valor total do item.

### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

...ao finalizar a compra o carrinho deve passar o valor total...

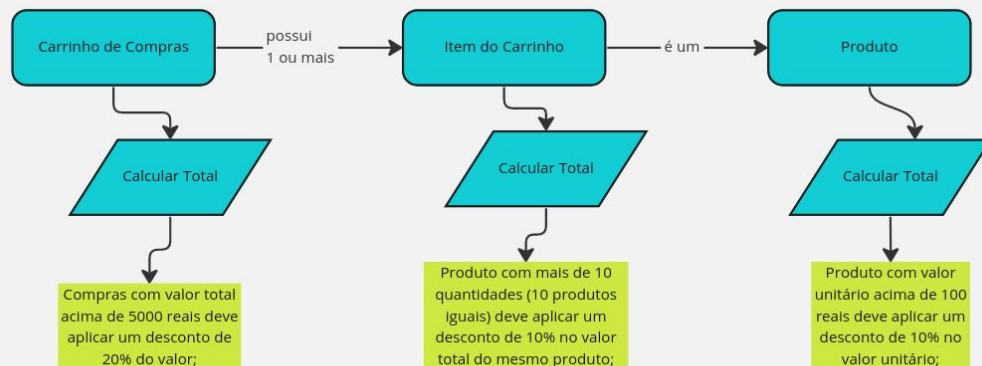
## Checkout

**Input:** Carrinho

**Output:** Valor total do carrinho

### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total



Todos os descontos podem ser acumulados.

# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?  
Como escolher um produto para adicionar no carrinho?

...onde é possível adicionar N produtos  
com N quantidades...

## Adicionar Item no Carrinho

**Input:** Produto, Quantidade.

**Output:** Valor total do item.

### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

...ao finalizar a compra o carrinho deve  
passar o valor total...

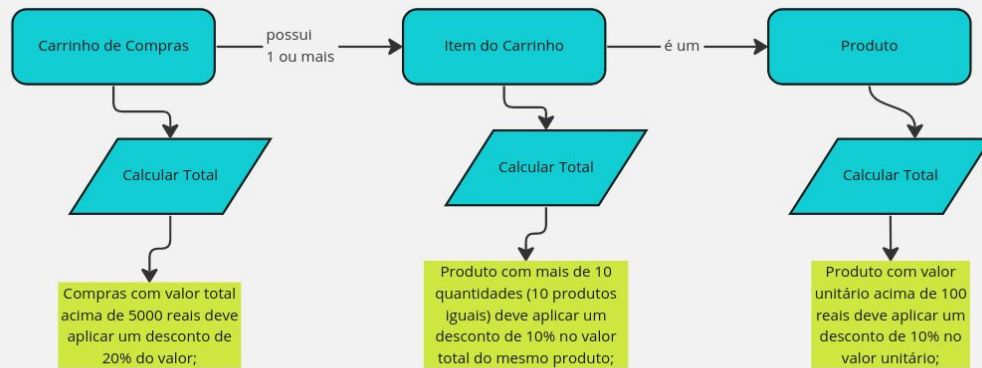
## Checkout

**Input:** Carrinho

**Output:** Valor total do carrinho

### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total



Todos os descontos podem ser acumulados.

# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?  
Como escolher um produto para adicionar no carrinho?  
Como criar um carrinho? Já é criado automaticamente?

...onde é possível adicionar N produtos com N quantidades...

...ao finalizar a compra o carrinho deve passar o valor total...

## Adicionar Item no Carrinho

**Input:** Produto, Quantidade.

**Output:** Valor total do item.

### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

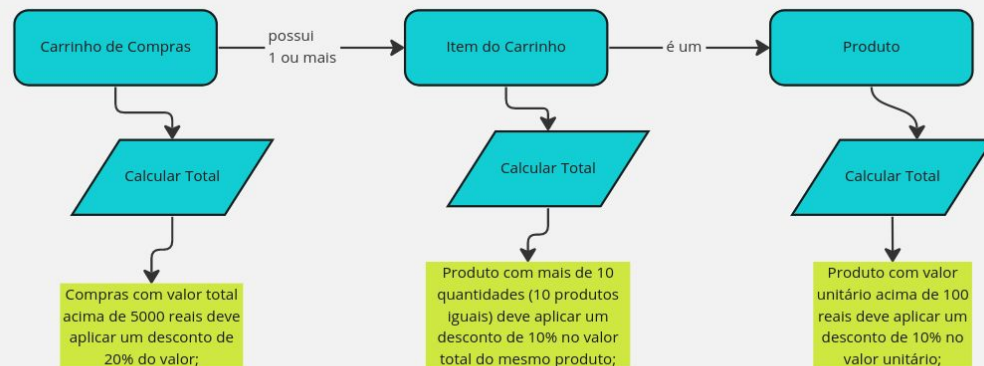
## Checkout

**Input:** Carrinho

**Output:** Valor total do carrinho

### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total



Todos os descontos podem ser acumulados.



# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?  
Como escolher um produto para adicionar no carrinho?  
Como criar um carrinho? Já é criado automaticamente?  
O carrinho é criado na primeira adição de item?

...onde é possível adicionar N produtos com N quantidades...

## Adicionar Item no Carrinho

**Input:** Produto, Quantidade.

**Output:** Valor total do item.

### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

...ao finalizar a compra o carrinho deve passar o valor total...

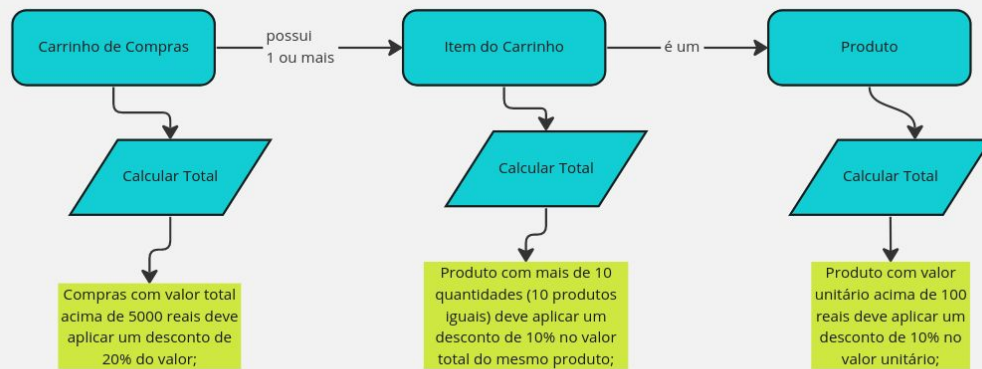
## Checkout

**Input:** Carrinho

**Output:** Valor total do carrinho

### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total





# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?  
 Como escolher um produto para adicionar no carrinho?  
 Como criar um carrinho? Já é criado automaticamente?  
 O carrinho é criado na primeira adição de item?  
 Posso excluir um produto?

...onde é possível adicionar N produtos com N quantidades...

...ao finalizar a compra o carrinho deve passar o valor total...

**Adicionar Item no Carrinho**

**Input:** Produto, Quantidade.  
**Output:** Valor total do item.

**Course:**

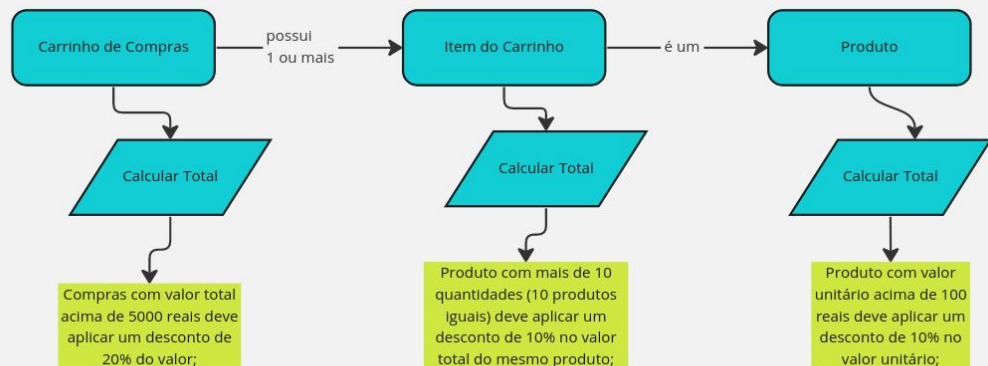
- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

**Checkout**

**Input:** Carrinho  
**Output:** Valor total do carrinho

**Course:**

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total



Todos os descontos podem ser acumulados.

# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?  
Como escolher um produto para adicionar no carrinho?  
Como criar um carrinho? Já é criado automaticamente?  
O carrinho é criado na primeira adição de item?  
Posso excluir um produto?  
E se o produto já existir no carrinho?

...onde é possível adicionar N produtos com N quantidades...

...ao finalizar a compra o carrinho deve passar o valor total...

## Adicionar Item no Carrinho

**Input:** Produto, Quantidade.

**Output:** Valor total do item.

### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

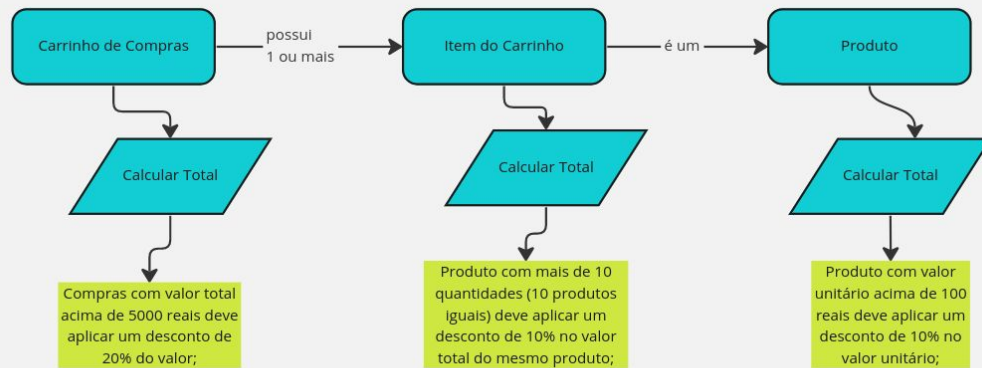
## Checkout

**Input:** Carrinho

**Output:** Valor total do carrinho

### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total



# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?  
Como escolher um produto para adicionar no carrinho?  
Como criar um carrinho? Já é criado automaticamente?  
O carrinho é criado na primeira adição de item?  
Posso excluir um produto?  
E se o produto já existir no carrinho?  
Consigo limpar/excluir o carrinho?

...onde é possível adicionar N produtos com N quantidades...

...ao finalizar a compra o carrinho deve passar o valor total...

## Adicionar Item no Carrinho

**Input:** Produto, Quantidade.

**Output:** Valor total do item.

### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

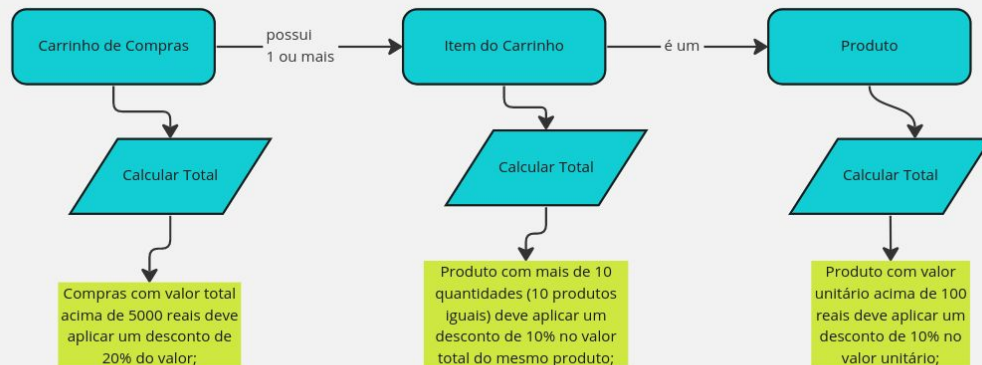
## Checkout

**Input:** Carrinho

**Output:** Valor total do carrinho

### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total



# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?  
Como escolher um produto para adicionar no carrinho?  
Como criar um carrinho? Já é criado automaticamente?  
O carrinho é criado na primeira adição de item?  
Posso excluir um produto?  
E se o produto já existir no carrinho?  
Consigo limpar/excluir o carrinho?  
etc...

...onde é possível adicionar N produtos com N quantidades...

...ao finalizar a compra o carrinho deve passar o valor total...

## Adicionar Item no Carrinho

**Input:** Produto, Quantidade.

**Output:** Valor total do item.

### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

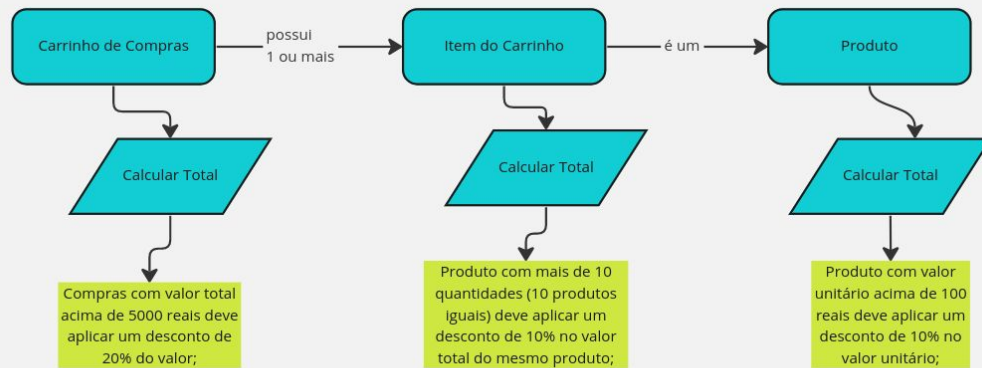
## Checkout

**Input:** Carrinho

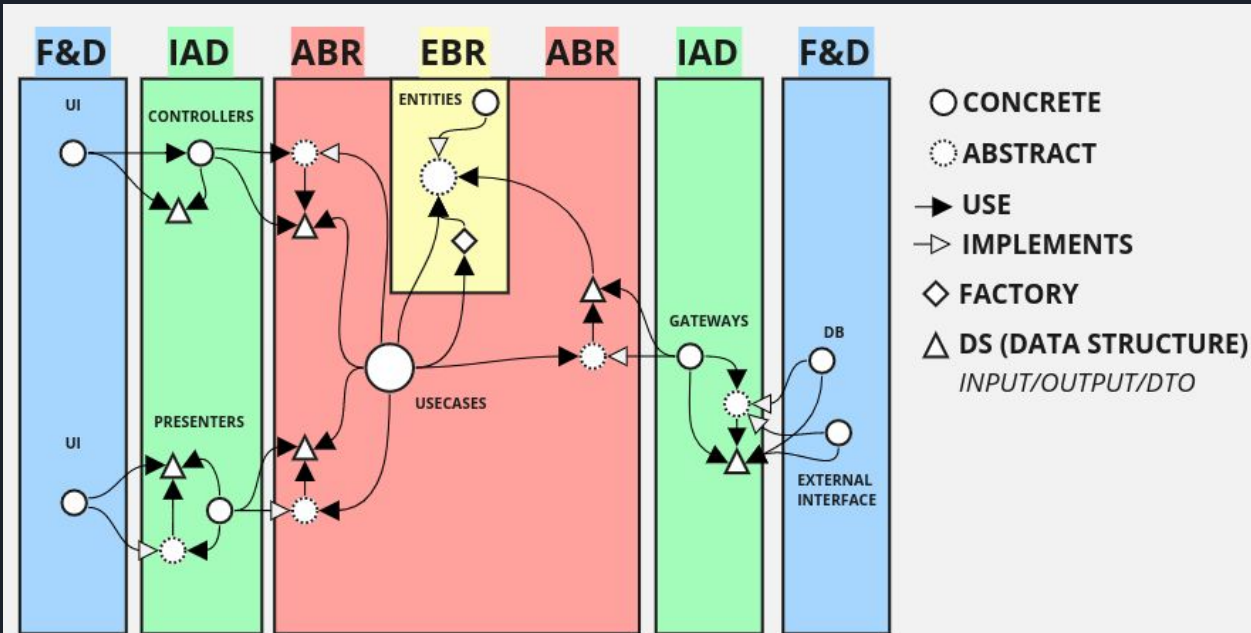
**Output:** Valor total do carrinho

### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total

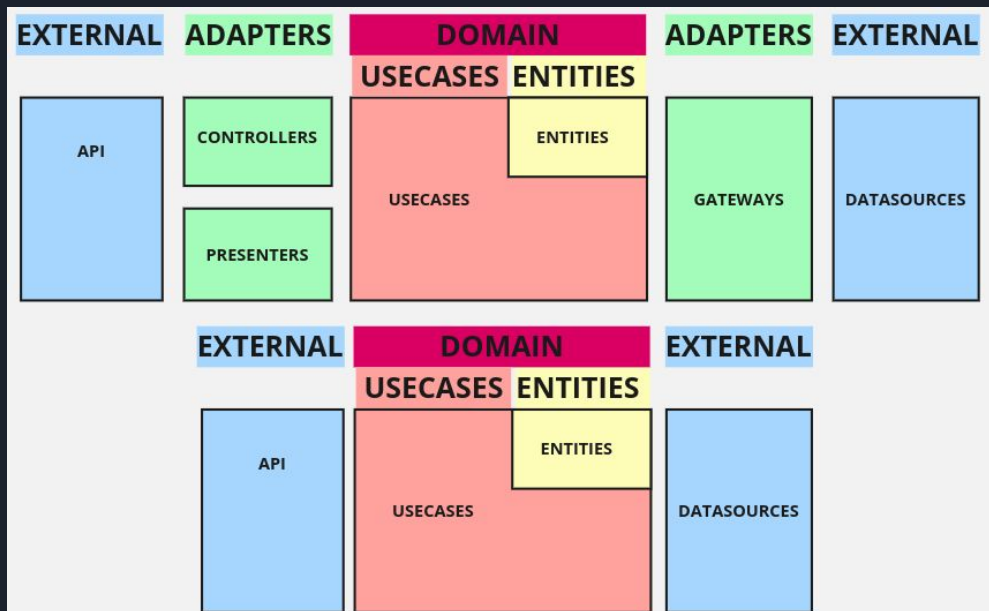


# Implementando Clean Arch

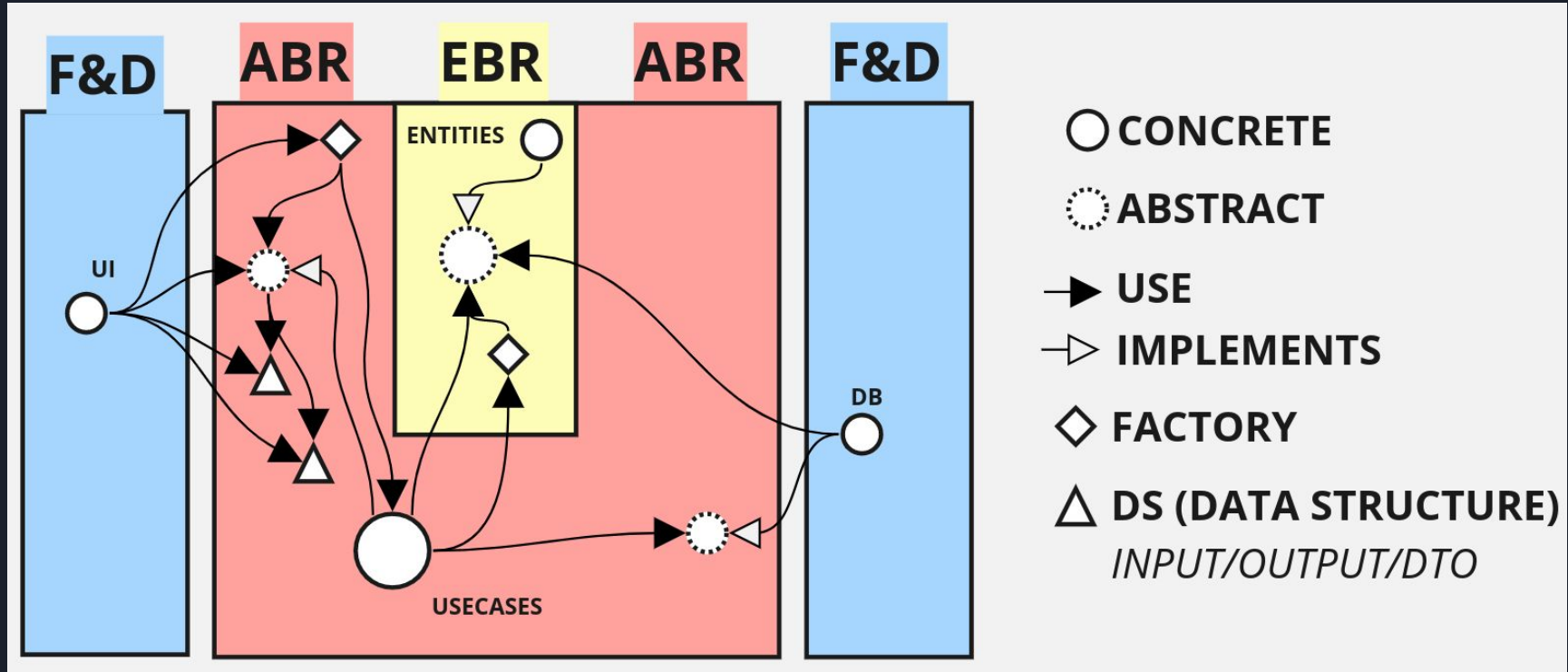


# Implementando Clean Arch

Um pouco de prática!

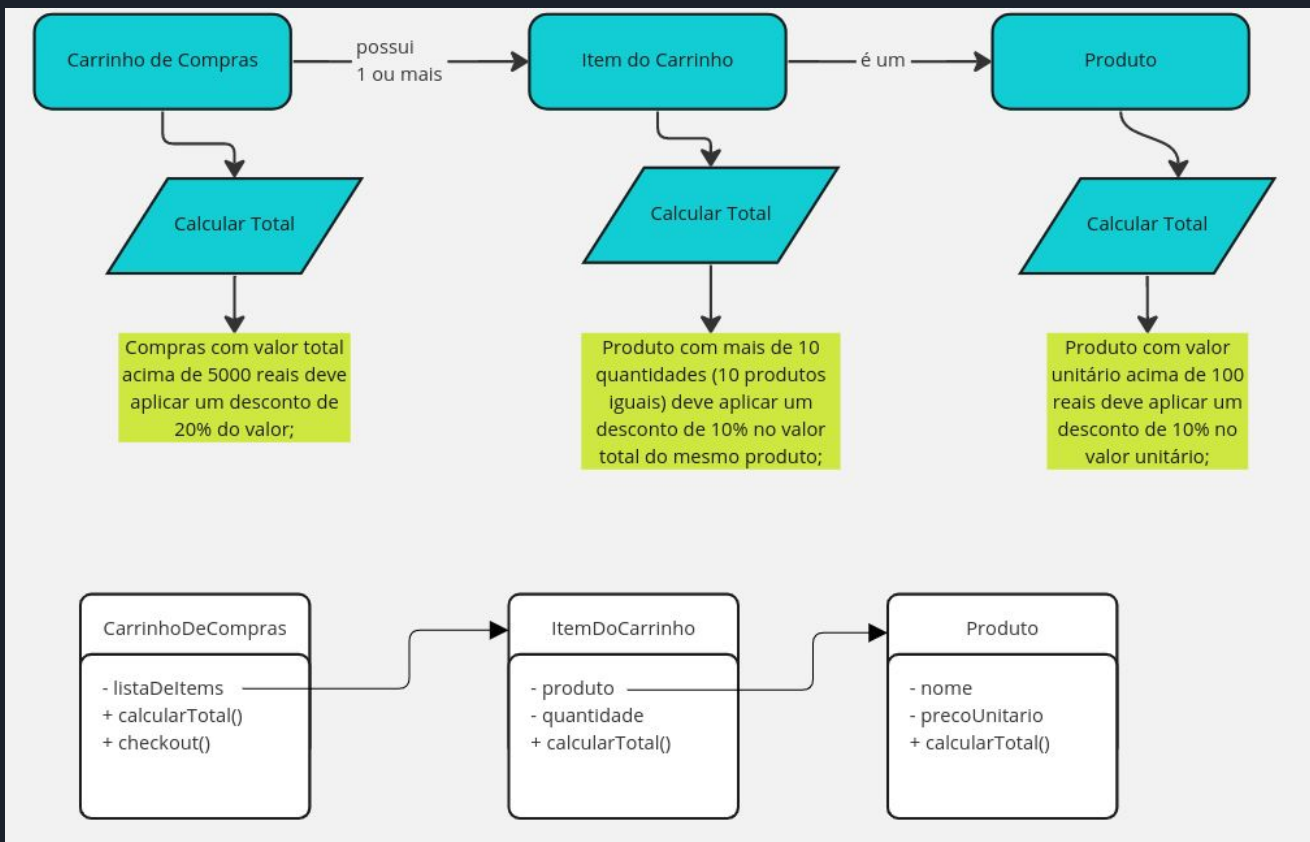


## Implementando Clean Arch





# Implementando Clean Arch





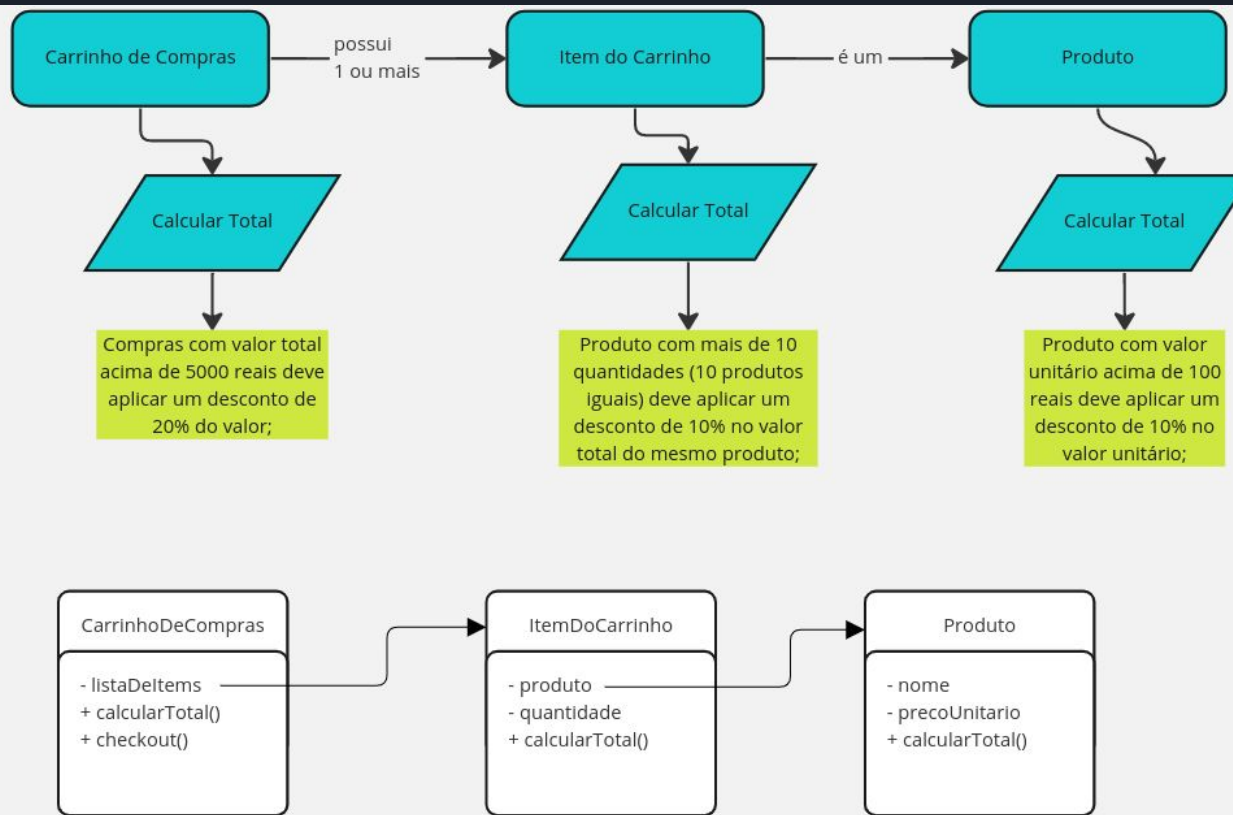


# Implementando Clean Arch

O que temos?

# Implementando Clean Arch

O que temos?



# Implementando Clean Arch

## O que temos?

...onde é possível adicionar N produtos com N quantidades...

### Adicionar Item no Carrinho

**Input:** Produto, Quantidade.  
**Output:** Valor total do item.

#### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

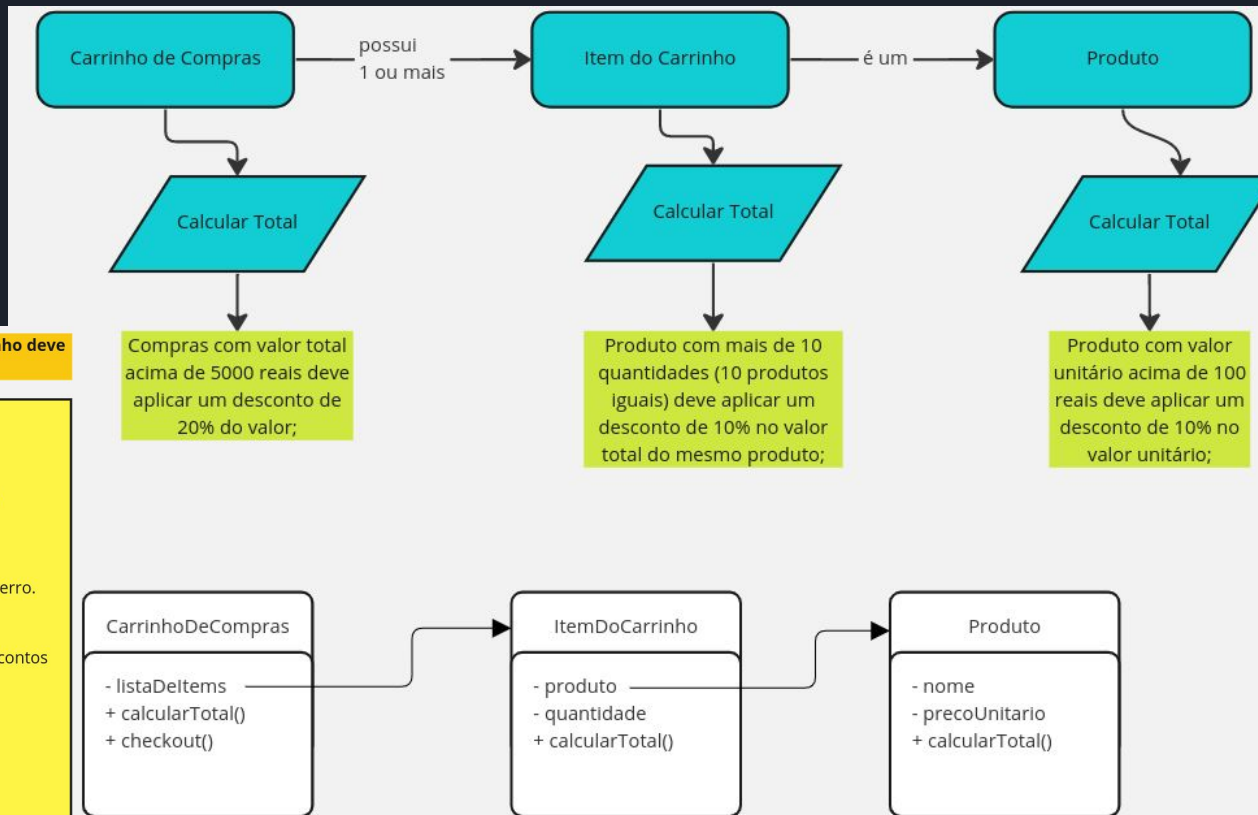
...ao finalizar a compra o carrinho deve passar o valor total...

### Checkout

**Input:** Carrinho  
**Output:** Valor total do carrinho

#### Course:

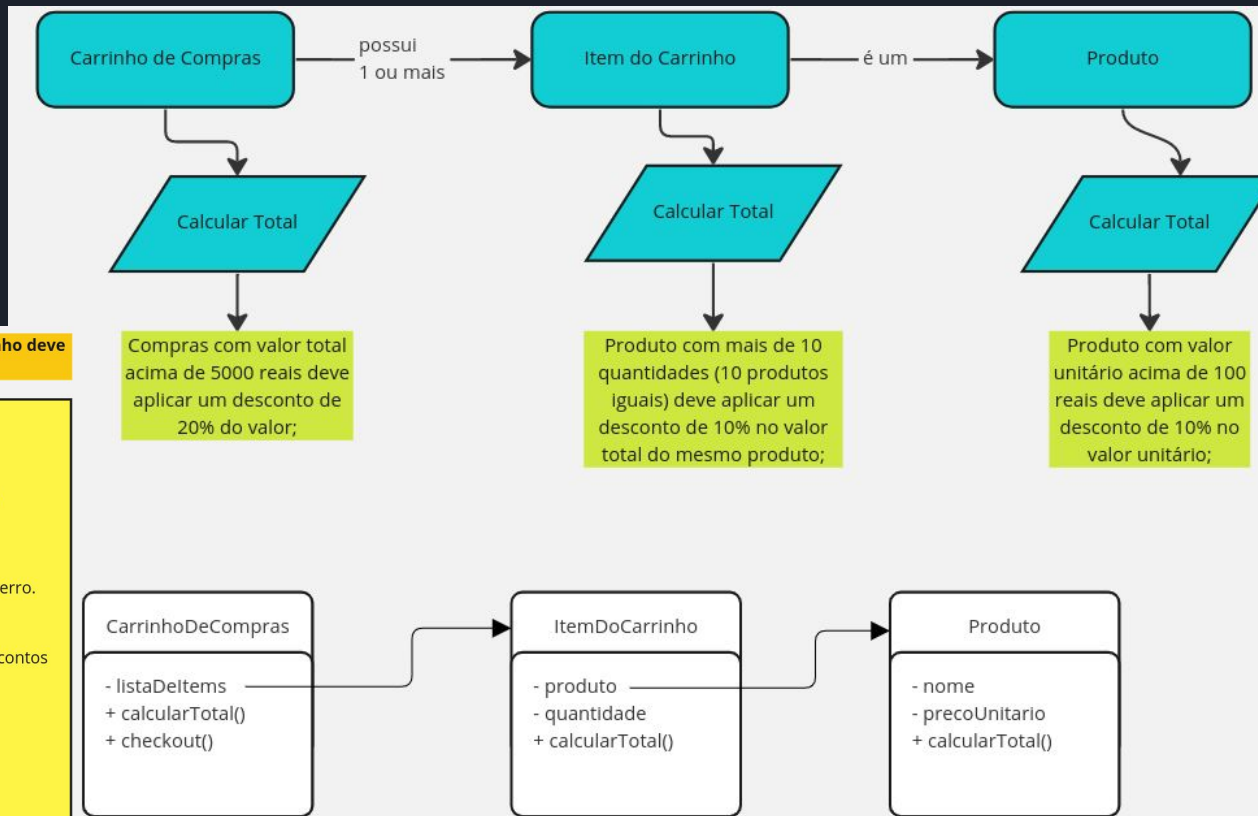
- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total



# Implementando Clean Arch

Qual o retorno de adicionar um item no carrinho?  
 Como escolher um produto para adicionar no carrinho?  
 Como criar um carrinho? Já é criado automaticamente?  
 O carrinho é criado na primeira adição de item?  
 Posso excluir um produto?  
 E se o produto já existir no carrinho?  
 Consigo limpar/excluir o carrinho?  
 etc...

## O que temos?



...onde é possível adicionar N produtos com N quantidades...

...ao finalizar a compra o carrinho deve passar o valor total...

### Adicionar Item no Carrinho

**Input:** Produto, Quantidade.  
**Output:** Valor total do item.

#### Course:

- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

### Checkout

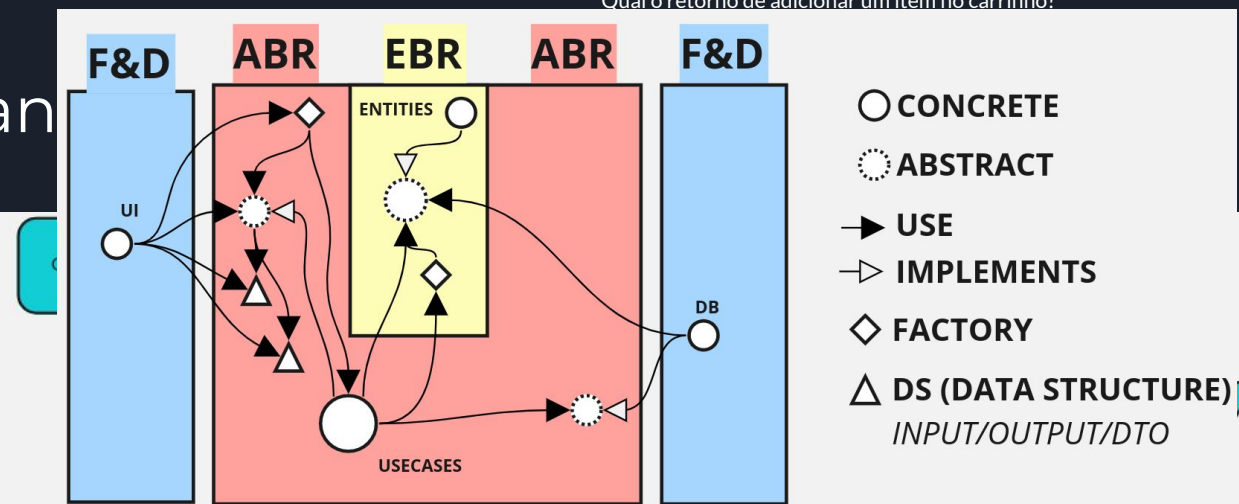
**Input:** Carrinho  
**Output:** Valor total do carrinho

#### Course:

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total

# Implementar

## O que temos?



...onde é possível adicionar N produtos com N quantidades...

...ao finalizar a compra o carrinho deve passar o valor total...

Compras com valor total acima de 5000 reais deve aplicar um desconto de 20% do valor;

Produto com mais de 10 quantidades (10 produtos iguais) deve aplicar um desconto de 10% no valor total do mesmo produto;

Produto com valor unitário acima de 100 reais deve aplicar um desconto de 10% no valor unitário;

**Adicionar Item no Carrinho**

**Input:** Produto, Quantidade.  
**Output:** Valor total do item.

**Course:**

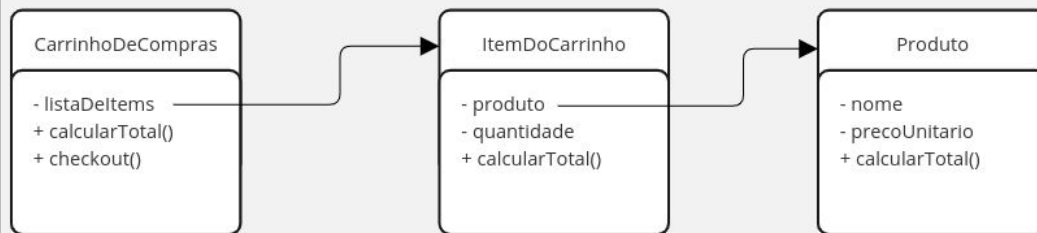
- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

**Checkout**

**Input:** Carrinho  
**Output:** Valor total do carrinho

**Course:**

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total

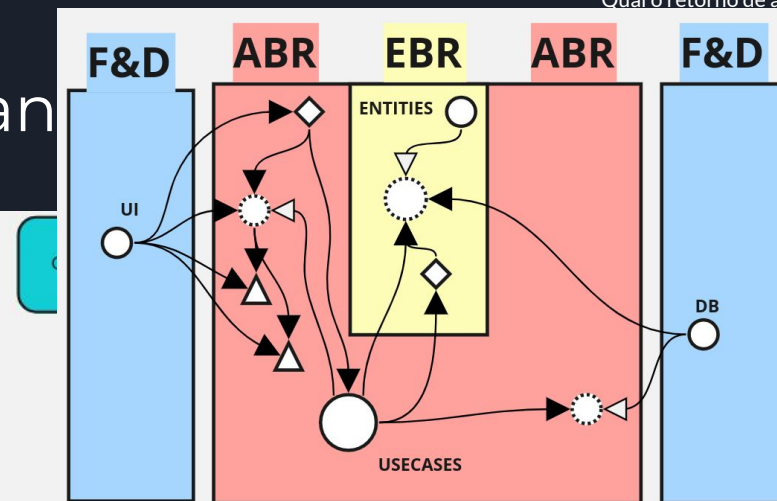


# Implementando

## Bora para o Código!

[https://github.com/brunotacca/secompp22\\_clean\\_arch\\_example](https://github.com/brunotacca/secompp22_clean_arch_example)

Qual o retorno de adicionar um item no carrinho?



○ CONCRETE

⊙ ABSTRACT

→ USE

▷ IMPLEMENTS

◇ FACTORY

△ DS (DATA STRUCTURE)  
INPUT/OUTPUT/DTO

...onde é possível adicionar N produtos com N quantidades...

...ao finalizar a compra o carrinho deve passar o valor total...

Compras com valor total acima de 5000 reais deve aplicar um desconto de 20% do valor;

Produto com mais de 10 quantidades (10 produtos iguais) deve aplicar um desconto de 10% no valor total do mesmo produto;

Produto com valor unitário acima de 100 reais deve aplicar um desconto de 10% no valor unitário;

**Adicionar Item no Carrinho**

**Input:** Produto, Quantidade.  
**Output:** Valor total do item.

**Course:**

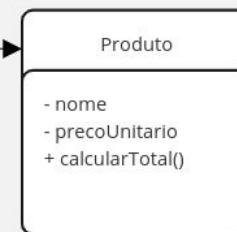
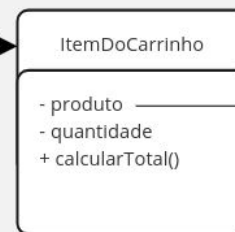
- Verificar se o produto existe.
- Mensagem de erro se não existir.
- Adicionar o item no carrinho.
- Calcular o valor total do item (com descontos)
- Retornar o preço total do item

**Checkout**

**Input:** Carrinho  
**Output:** Valor total do carrinho

**Course:**

- Verificar se carrinho existe
- Se não existir, mensagem de erro.
- Finalizar o carrinho
- Calcular o valor total (com descontos acumulados).
- Retornar o valor total





SEC**U**MPP22

# Clean Architecture

**Bruno Tacca**  
Ciência da Computação - UNESP

**Obrigado!**