# Separation of Concerns e Factory Pattern

# 1.
# Separation of Concerns

Separação de preocupações e modularização

# Introdução

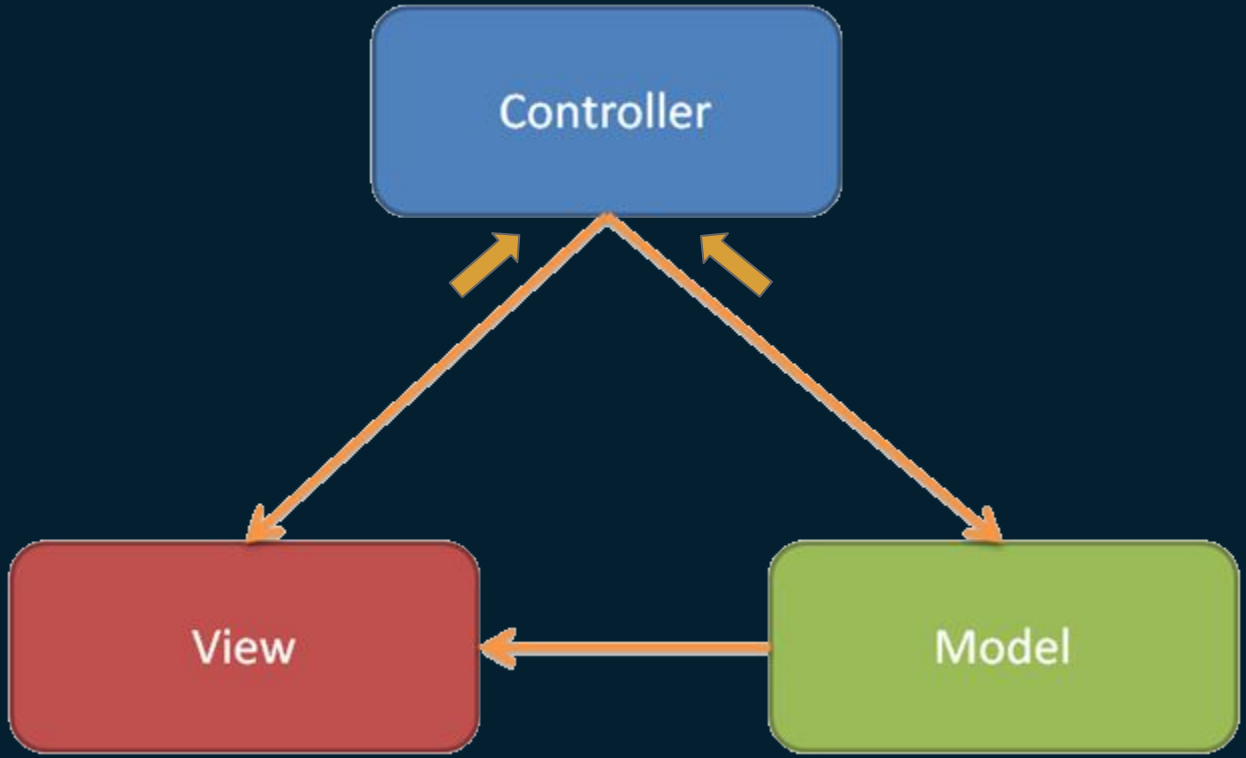
- Softwares com baixa manutenibilidade
- Softwares de alta complexidade

*"A manutenção de software pode tomar 80% do tempo total no desenvolvimento do produto de software."* - International Congress of Administration (2010)

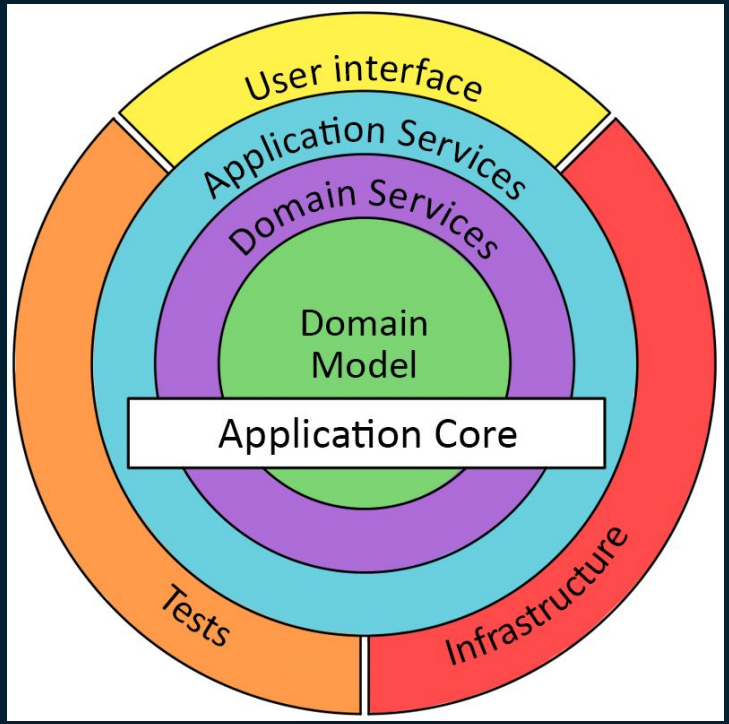
# Conceitos

- Modularização do código
- Responsabilidade Única

# MVC (Model-View-Controller)



# Onion architecture



# 2.

# Factory Method

Padrão para criação de objetos

# Padrões de Criação

- “Creational design patterns abstract the instantiation process.” - Elements of Reusable Oriented Software

- Abstract Factory
- Factory Method
- Builder
- Singleton
- Prototype

# Factory Method - Como funciona?

- “Um padrão que define uma interface para criar um objeto, mas permite às classes decidirem qual classe instanciar. O Factory Method permite a uma classe deferir a instanciação para subclasses.” - Gang of Four

- Interface ou classe abstrata
- Subclasses

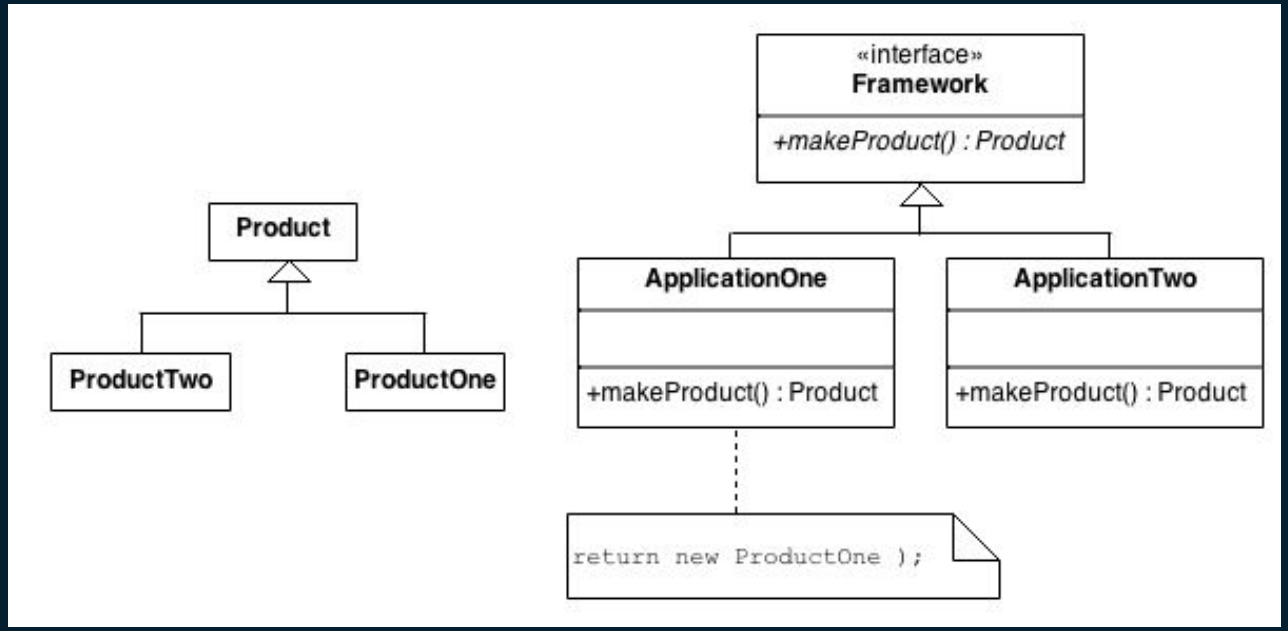
# Exemplo

```
public interface Blacksmith {
  Weapon manufactureWeapon(WeaponType weaponType);
}

public class ElfBlacksmith implements Blacksmith {
  public Weapon manufactureWeapon(WeaponType weaponType) {
    return new ElfWeapon(weaponType);
  }
}

public class OrcBlacksmith implements Blacksmith {
  public Weapon manufactureWeapon(WeaponType weaponType) {
    return new OrcWeapon(weaponType);
  }
}
```

# UML


«interface»
Framework
+makeProduct() : Product
Product
ProductTwo
ProductOne
ApplicationOne
+makeProduct() : Product
ApplicationTwo
+makeProduct() : Product
return new ProductOne );

# Vantagens

- Baixo Acoplamento
- Leitura mais fácil
- Limpa “sujeira” do cliente
- Separação de responsabilidade

# Patternite

- Quero usar != Necessidade de uso
- Ambientes
- Tamanhos de projeto

# Referências


- http://java-design-patterns.com
- http://coding-geek.com/design-pattern-factory-patterns/
- https://sourcemaking.com/design_patterns/factory_method
- Elements of Reusable Object-Oriented Software - [Gamma, E. - et. al]