



CES/JF

*Centro de Ensino Superior
de Juiz de Fora*

Padrões de Projeto (Design Patterns)

Prof. MSc. Tassio Sirqueira

tassio@tassio.eti.br

O que é padrão de projeto?

- Padrão de projeto é a tradução de Design Pattern, e tem como objetivo resolver problemas que ocorrem com frequência dentro da orientação a objetos.
- Nenhum padrão de projeto é um esqueleto pronto e definitivo de solução para um problema do dia a dia, ele é um modelo.
- Os padrões podem ser estruturados grosseiramente em tres categorias diferentes.
 - Criacional;
 - Estrutural;
 - Comportamental;

O que é GoF?

- GoF ou Gang of Four (Gangue dos quatro).
- A GoF é formada por Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, que juntos escreveram um livro chamado [Design Patterns: Elements of Reusable Object-Oriented Software](#) em 1994. O livro ganhou muita popularidade e inspirou muitos outros depois.
- Eles estabeleceram 23* padrões de projeto separados em 3 categorias, padrões de criação, estruturais e comportamentais.
- * Padrões definidos originalmente pela GoF.

Padrões de Criação

- Focados na criação de objetos
 - ✓ Abstract Factory*
 - ✓ Builder*
 - ✓ Factory Method*
 - ✓ Multiton (Considerado um anti-pattern!)
 - ✓ Pool
 - ✓ Prototype*
 - ✓ SimpleFactory
 - ✓ Singleton* (Considerado um anti-pattern!)
 - ✓ StaticFactory

Padrões Estruturais

- Focados na associação entre objetos
 - ✓ Adapter*
 - ✓ Bridge*
 - ✓ Composite*
 - ✓ DataMapper
 - ✓ Decorator*
 - ✓ DependencyInjection
 - ✓ Facade*
 - ✓ FluentInterface
 - ✓ Flyweight*
 - ✓ Proxy*
 - ✓ Registry

Padrões Comportamentais

- Focados nas interações entre objetos
 - ✓ Chain Of Responsibilities*
 - ✓ Command*
 - ✓ Interpreter*
 - ✓ Iterator*
 - ✓ Mediator*
 - ✓ Memento*
 - ✓ NullObject
 - ✓ Observer*
 - ✓ Specification
 - ✓ State*
 - ✓ Strategy*
 - ✓ TemplateMethod*
 - ✓ Visitor*

Outros

- A engenharia de software está sempre em busca de novas soluções para deixar os software melhores e com isso a descoberta de novos padrões ainda estão surgindo, como, por exemplo:
 - ✓ Delegation
 - ✓ ServiceLocator
 - ✓ Repository
 - ✓ EAV

Exemplos de Implementação

- **Design Patterns em PHP:**
 - <https://github.com/domnikl/DesignPatternsPHP>
- **Design Patterns em Java:**
 - <https://github.com/MarcosX/Padr-es-de-Projeto>

Padrão Arquitetural MVC

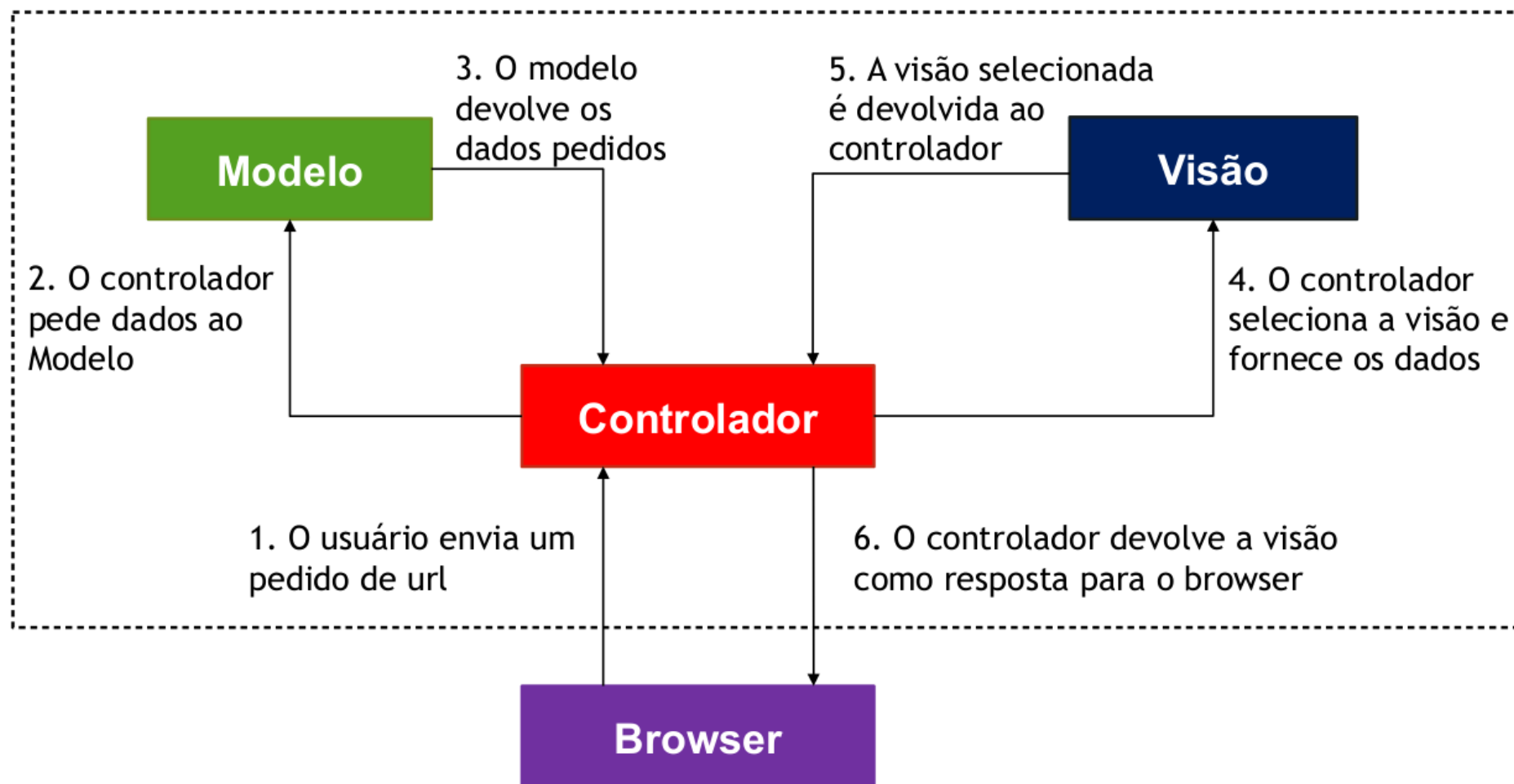
- **Model-View-Controller (MVC)**
- É um padrão de arquitetura de aplicações que divide a aplicação em três camadas: a visão (view), o modelo (model), e o controlador (controller).
- O padrão MVC foi desenvolvido em 1979 por Trygve Reenskaug, com a finalidade de ser utilizado como arquitetura para aplicativos desktop. Entretanto, o padrão se popularizou para uso em sistemas web, a partir da adesão de milhares de Frameworks de mercado.

O que é Arquitetura de Aplicação?

- A arquitetura de uma aplicação é a definição de suas estruturas.
- Dividir a aplicação em camadas: uma de interface do usuário denominada *View*, uma para manipulação lógica de dados, chamada *Model*, e uma terceira camada de fluxo da aplicação, chamada *Control*)
- Cria a possibilidade de exibir uma mesma lógica de negócios através de várias interfaces.
- Isolar a camada de negócios (*Model*) das demais camadas do sistema, de forma a facilitar a sustentabilidade do código.
- A implementação do controlador deve permitir que esta camada receba os eventos da interface e os converta em ações no modelo.



Modelo MVC



Padrão DAO (Data Access Object)

- O padrão DAO é um padrão de projeto que abstrai e encapsula os mecanismos de acesso a dados, escondendo os detalhes da execução da origem dos dados.
- O padrão de projeto DAO surgiu com a necessidade de separarmos a lógica de negócios da lógica de persistência de dados.
- Este padrão permite que possamos mudar a forma de persistência sem que isso influencie em nada na lógica de negócio, além de tornar nossas classes mais legíveis.

Padrão DAO (Data Access Object)

- Este padrão permite criar as classes de dados independentemente da fonte de dados ser um BD relacional, um arquivo texto, um arquivo XML, JSON e etc.
- Para isso, ele encapsula os mecanismos de acesso a dados e cria uma interface de cliente genérica, para fazer o acesso aos dados, permitindo que os mecanismos de acesso sejam alterados, independentemente do código que utiliza os dados.