

# Programação Web

## Módulo 6 - Injeção de dependência e anotações Spring

Daniel Lucrédio  
daniel.lucradio@ufscar.br

Este conteúdo foi baseado no conjunto de slides gentilmente cedido pelo prof. Delano Medeiros Beder  
(<https://github.com/delanobeder/DSW1>)

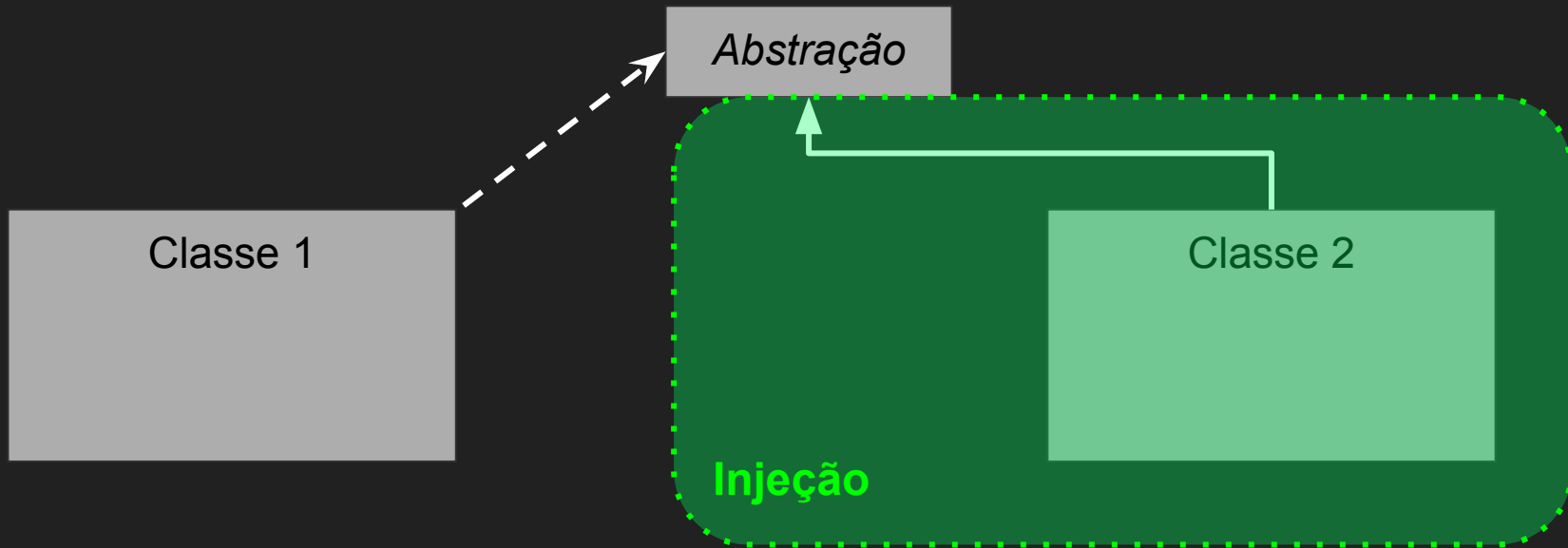
# Injeção de Dependência

O Spring **nasceu** como um framework para injeção de dependência



# Injeção de Dependência

O Spring **nasceu** como um framework para injeção de dependência



# Exemplo de Injeção de Dependência

Demonstração 20

# Anotações Spring

# Anotações básicas - em nível de classe

**@Controller** - uma classe é um controlador MVC

**@Service** - uma classe encapsula regras de negócio

**@Repository** - uma classe cuida da persistência

**@Component** - uma classe gerenciada pelo Spring

# @Autowired

- Atributo, Construtor ou método "setter"
- Spring vai encontrar a classe e instanciá-la

@Autowired

Demonstração 21



# @Configuration / @Bean / @Resource

Permite configurar instâncias específicas

**@Configuration** - Anotação de classe que possibilita registrar beans

**@Bean** - Anotação de método que possibilita instanciar beans

**@Resource** - Parecido com @Autowired, porém aqui a busca é pelo nome, e não pelo tipo

@Configuration / @Bean / @Resource

Demonstração 22

# Anotações específicas do Spring MVC controller

**@RequestMapping** (e @GetMapping, @PostMapping, @DeleteMapping, @PutMapping...) - Anotação de método para tratar uma requisição HTTP

**@PathVariable** - Usado junto com um request mapping, para identificar parâmetros passados dentro do caminho (path)

**@RequestParam** - Usado junto com um request mapping para identificar parâmetros passados na query (url?nome=valor)

**@ModelAttribute** - Permite configurar um atributo do modelo antes de qualquer tratador de requisição

**@Valid / BindingResult** - Injeção de validação no back-end

# Anotações específicas do Spring MVC controller

Demonstração 23

# Vamos migrar a aplicação JSP + Servlets para Spring MVC

Demonstração 24

# Comparando - classes Java

- O trabalho de **conversão e validação** é automático
  - **Antes** estava nos servlets + bean específico para form
  - **Agora** ficou embutido nos componentes Spring
- A **navegação** ficou mais simples
  - **Antes** estava explícito com forward nos servlets
  - **Agora** está no retorno dos métodos de ação e gerenciado pelo controlador

# Comparando - views

Página	JSP	ThymeLeaf
index		+ Exibição de mensagens + Gerenciamento de sessão fora da visão
palpiteForm		+ Exibição de mensagens + Amarração com modelo
confirmarPalpite		+ Gerenciamento de sessão fora da visão
listaPalpites		+ Exibição de dados

# Spring MVC = melhor separação de interesses

Visão - somente apresentação

Controlador - gestão de sessão, lógica de controle

Modelo - quase a mesma coisa, exceto pela facilidade de injetar os DAOs e conexão com o banco de dados



# Último exercício

Demonstração 25

Fim