

# Introdução ao Spring Boot MVC no Java Servlet Stack com a aplicação Pet Clinic

marco.mangan@gmail.com

2019-06-15

1 Visão Geral

2 Spring

3 Pet Clinic

# Introdução ao Spring Boot MVC no Java Servlet Stack com a aplicação Pet Clinic

Esta apresentação baseia-se nos guias divulgados pela Pivotal a respeito da programação no *framework* Spring [3] e da aplicação Pet Clinic [4] e no material de Thomas Woehlke a respeito da variante da Pet Clinic [5] para o *framework* Java Enterprise Edition [2]. Foi também utilizado material elaborado pela Data Flair [1].

Uma aplicação Web é um sistema distribuído de alta complexidade. Diversos *frameworks* de aplicação foram propostos para simplificar a tarefa de desenvolver esse tipo de aplicação.

O Spring é um dos *frameworks* mais influentes dentro e fora do ecossistema Java. Sua implementação em código aberto adota conceitos inovadores como programação orientada a aspectos, programação por convenção e injeção de dependências. Além disso, o *framework* incentiva e divulga práticas como teste e compilação automáticos.

A aplicação Spring Pet Clinic [4] é um arquétipo que apresenta diversas soluções para compor uma aplicação Web contemporânea. A palestra é dirigida para iniciantes. Serão apresentados conceitos de programação e comentados guias selecionados disponíveis no site da spring.io [3]. Ao final da palestra o participante deve ser capaz de compreender, implantar e alterar o código da aplicação Pet Clinic.

- ✓ Distribuições Linux e *Aplicações Livres*
  - Administração de Sistemas, Redes e *Segurança*
- ✓ *Desenvolvimento*, Banco de Dados e Web
  - Virtualização, *Big Data* e *Cloud Computing*
  - Kernel, *Sistemas Embarcados* e *Internet das Coisas*
  - *Empreendedorismo* e *Negócios*
  - Comunidade e *Filosofia*

- Uma aplicação Web (*Web application*) é um sistema distribuído de alta complexidade.
- Um sistema distribuído apresenta problemas com comunicação, segurança, consistência e correção.
- Uma arquitetura cliente e servidor é a situação mais simples de uma aplicação distribuída.
- A World Wide Web é composta por três elementos: URL, HTTP e HTML.
- A Web não foi proposta para permitir aplicações.

# Breve histórico de aplicações distribuídas

- 1970s Internet, Sockets
- 1980s SGBD, RPC, OMG CORBA
- 1990s W3C World Wide Web, HTML, HTTP, URL
- 2000s Serviços REST e SOAP
- 2010s Aplicativos para iOS e Android, IoT, Nuvem

- A programação orientada a aspectos permite o tratamento de preocupações ortogonais, como autorização, registros de auditoria e persistência.
- Um pós-compilador insere instruções, sem que o programador necessita escrever código repetitivo ou relacionado com uma tecnologia específica.
- Exemplo: código CRUDL do mapeamento objeto-relacional.



# *Convention over configuration*

- Programar é realizar escolhas. O conjunto de escolhas é referido como arquitetura.
- Em diversos sistemas, as escolhas recorrentes se tornam uma convenção. O programador não necessita informar configurações que são determinadas pela convenção.
- Exemplo: nomes de pastas, nomes de classes.

- A injeção de dependências simplifica testes e a composição de aplicações.
- O ambiente de execução providencia as instâncias e as atribui aos campos de objetos da aplicação em execução.
- Exemplo: referência ao repositório de dados.

- A inversão de controle reduz o consumo de recursos e reduz dependências.
- Princípio de Hollywood.
- Exemplo: seu controlador é ativado pelo DispatcherServlet.

- Web, CERN, Tim Berners-Lee (1989)  
<https://www.w3.org/History/1989/proposal.html>
- AOP, PARC, Gregor Kiczales (1997)  
[https://www.researchgate.net/profile/Gregor\\_Kiczales/publication/4200498\\_Aspect-oriented\\_programming\\_and\\_modular\\_reasoning/links/0912f50c81e2e945f6000000/Aspect-oriented-programming-and-modular-reasoning.pdf](https://www.researchgate.net/profile/Gregor_Kiczales/publication/4200498_Aspect-oriented_programming_and_modular_reasoning/links/0912f50c81e2e945f6000000/Aspect-oriented-programming-and-modular-reasoning.pdf)
- REST, UCI, Roy Fielding (2000) [https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf)

# Breve histórico da plataforma Java

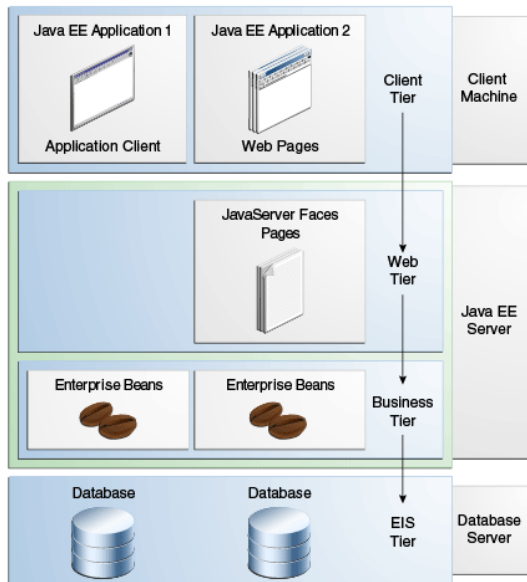
Version	Release date	End of Public Updates <sup>[5]</sup>	Extended Support Until
JDK Beta	1995	?	?
JDK 1.0	January 1996	?	?
JDK 1.1	February 1997	?	?
J2SE 1.2	December 1998	?	?
J2SE 1.3	May 2000	?	?
J2SE 1.4	February 2002	October 2008	February 2013
J2SE 5.0	September 2004	November 2009	April 2015
Java SE 6	December 2006	April 2013	December 2018
Java SE 7	July 2011	April 2015	July 2022
Java SE 8 (LTS)	March 2014	January 2019 (commercial) December 2020 (non-commercial)	March 2025
Java SE 9	September 2017	March 2018	N/A
<b>Java SE 10 (18.3)</b>	March 2018	September 2018	N/A
Java SE 11 (18.9 LTS)	September 2018	March 2019 from Oracle Later from OpenJDK	Vendor specific
Java SE 12 (19.3)	March 2019	September 2019	N/A
<b>Legend:</b> <span style="color: red;">■</span> Old version <span style="color: yellow;">■</span> Older version, still supported <span style="color: green;">■</span> Latest version <span style="color: lightblue;">■</span> Future release			

, Inc. [US] | <https://github.com/openjdk/jdk/blob/master/src/java.base/share/classes/java/lang/String.java>

```
117 * or {@code java.lang.invoke.StringConcatFactory} depending on the JDK version. The
118 * implementation of string conversion is typically through the method {@code toString},
119 * defined by {@code Object} and inherited by all classes in Java.
120 *
121 * @author Lee Boynton
122 * @author Arthur van Hoff
123 * @author Martin Buchholz
124 * @author Ulf Zibis
125 * @see java.lang.Object#toString()
126 * @see java.lang.StringBuffer
127 * @see java.lang.StringBuilder
128 * @see java.nio.charset.Charset
129 * @since 1.0
130 * @jls 15.18.1 String Concatenation Operator +
131 */
132
133 public final class String
134     implements java.io.Serializable, Comparable<String>, CharSequence,
135         Constable, ConstantDesc {
136
```

● Captura Retangular

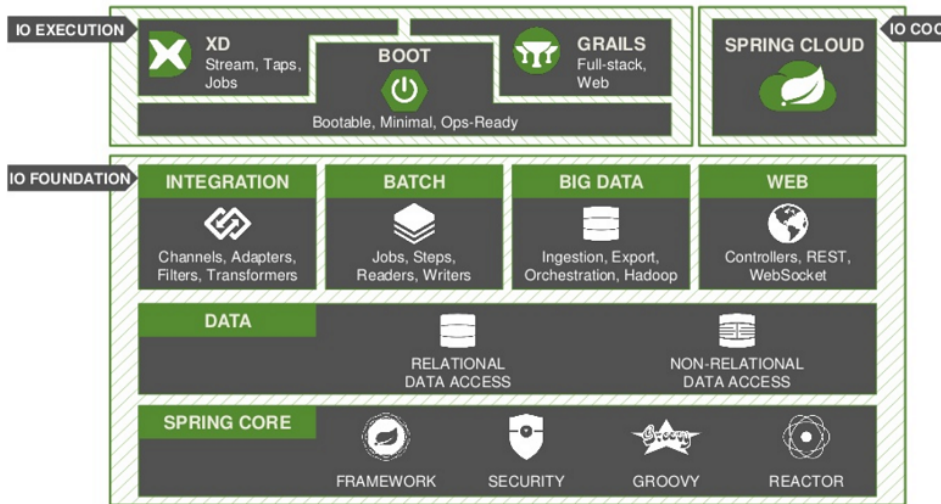
# Java EE (1998)



- O Spring apresenta diversos componentes.
- Os mais importantes para o iniciante são: Boot, Data e MVC.

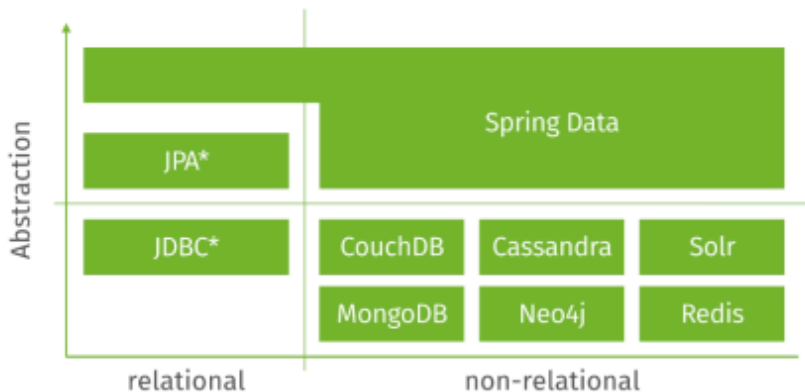


# Spring IO Platform

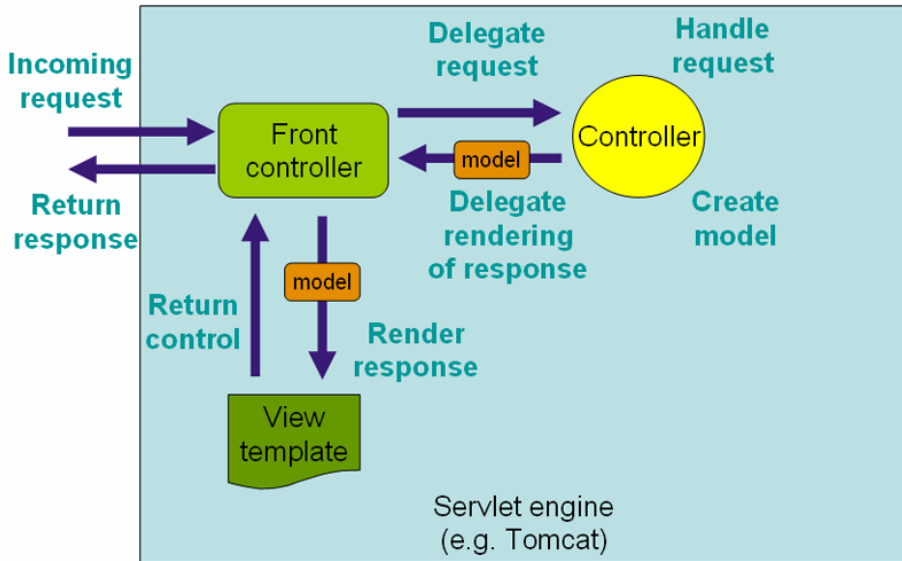


- Command Line Interface (CLI)
- Starters
- Autoconfigure
- Actuator
- Ferramentas (Tools)
- Amostras (Samples)

## Spring Data

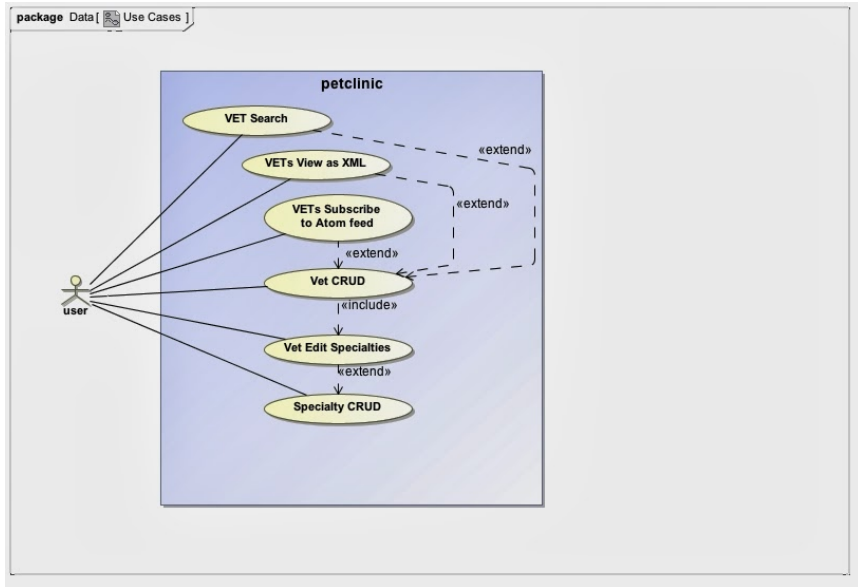


# Spring MVC

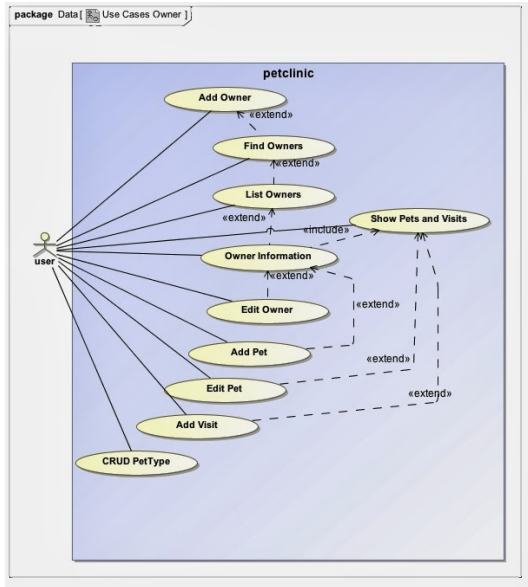


- A aplicação Spring Pet Clinic é um arquétipo que apresenta diversas soluções para compor uma aplicação Web contemporânea.

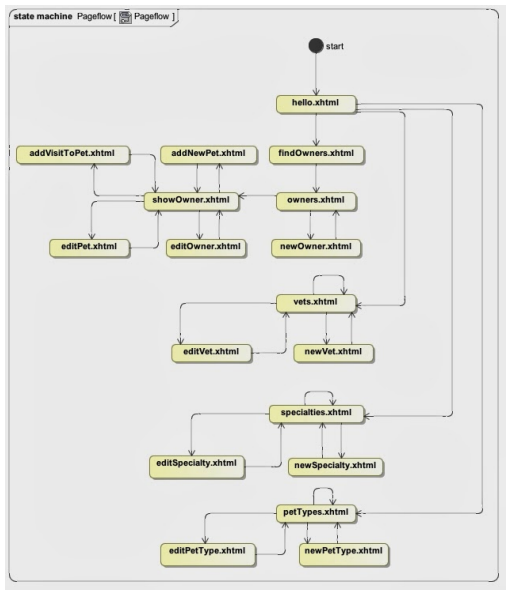
# Use Cases (I)



# Use Cases (II)

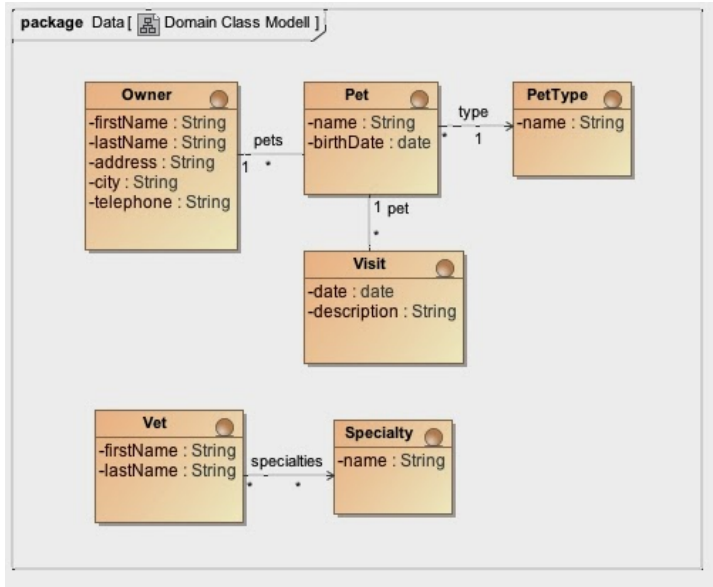


# Page Flow





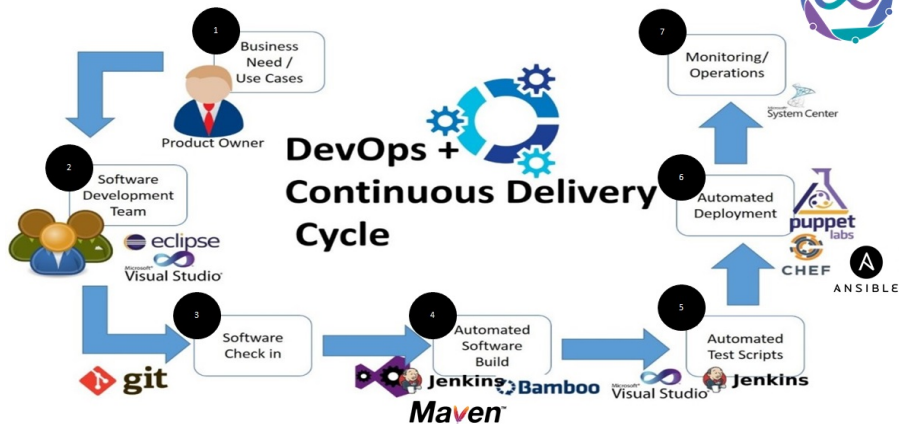
# Application Domain



# E agora?

- Terminaste?
- Podemos ver algum código?
- Podemos programar agora?






# DevOps Life Cycle



# O que faço agora?

- instale e aprenda as ferramentas: terminal, Git, GitHub, Travis, Sonar
- acompanhe os guias da Spring.io
- clone e execute a Spring Pet Clinic (SPC)
- altere a SPC para se tornar o SEU projeto

- Building Java Projects with Maven
- Building an Application with Spring Boot
- Serving Web Content with Spring MVC
- Accessing Relational Data using JDBC with Spring
- Accessing Data with JPA
- Validating Form Input
- Testing the Web Layer
- Securing a Web Application
- Managing Transactions

-  [Data Flair](#). Spring Framework Tutorial.
-  [Oracle](#). Java EE 7 Tutorial.
-  [Spring](#). Guides.
-  [Spring](#). Petclinic.
-  [Thomas Woehlke](#). Java EE 7 Pet Clinic.