



INSTITUTO FEDERAL  
Rio Grande do Sul

# Desenvolvimento de Sistemas I

*Spring Framework*

**Professor:** Jezer Machado de Oliveira



# Framework

---

Framework é o esqueleto-base sobre o qual uma aplicação é construída, constituída de uma estrutura de classes com implementações incompletas, que estendidas, permitem produzir novos aplicativos. A grande vantagem desta abordagem é a reutilização de código e projeto, que tem por intuito diminuir o tempo e o esforço no desenvolvimento de softwares.



# Framework x Biblioteca

---

- **Biblioteca**

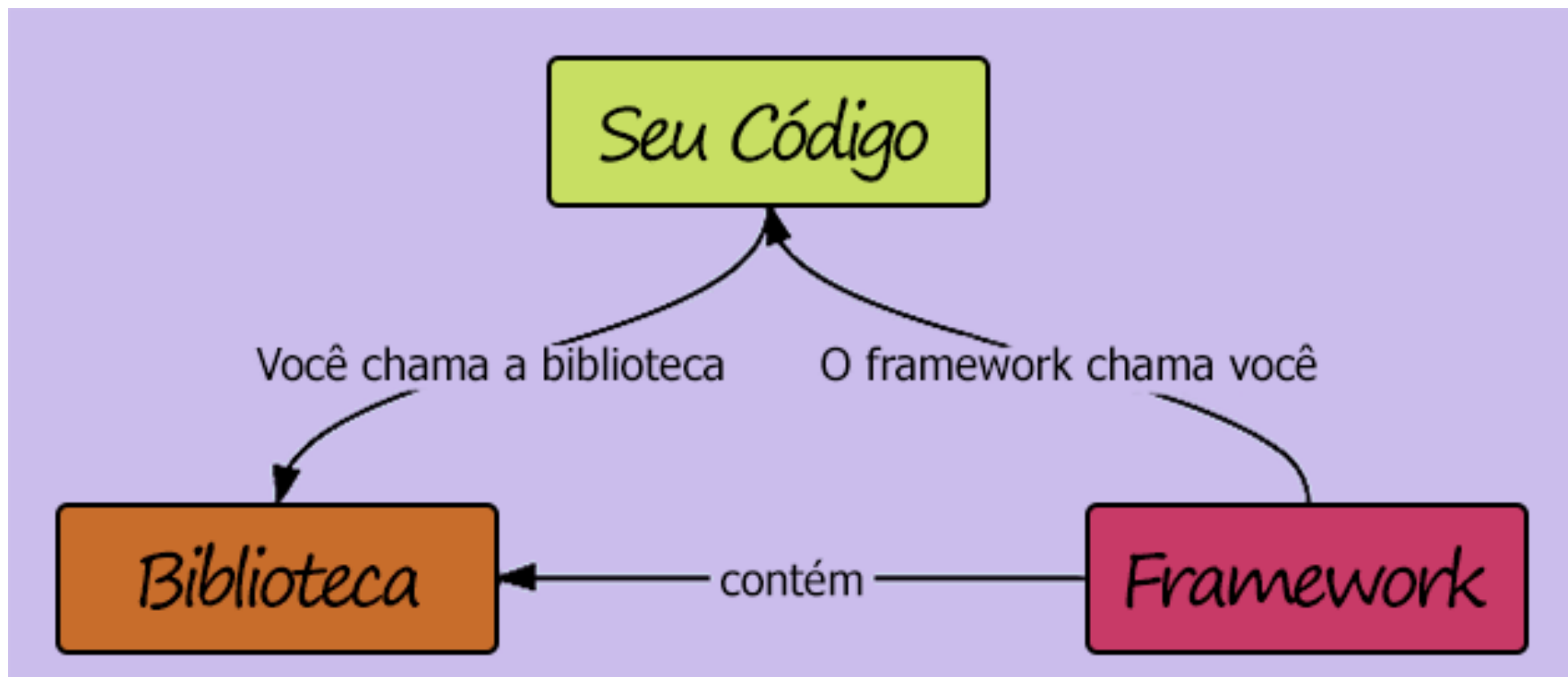
- Conjunto de funções para realizar determinada tarefa
- Controle continua com o programa principal

- **Framework**

- Conjunto de bibliotecas que desempenham uma gama de funções
- O controle do programa passa para o framework



# Framework x Biblioteca



# Framework - Prós

---

- **Eficiência**
- **Reuso**
- **Segurança**
- **Documentação**
- **Padrões de codificação**



# Framework - Contrás

---

- **Códigos desnecessários**
- **Dificuldade de customização**
- **Dependência**
- **Não tem a mesma vida útil de uma linguagem**



# Spring

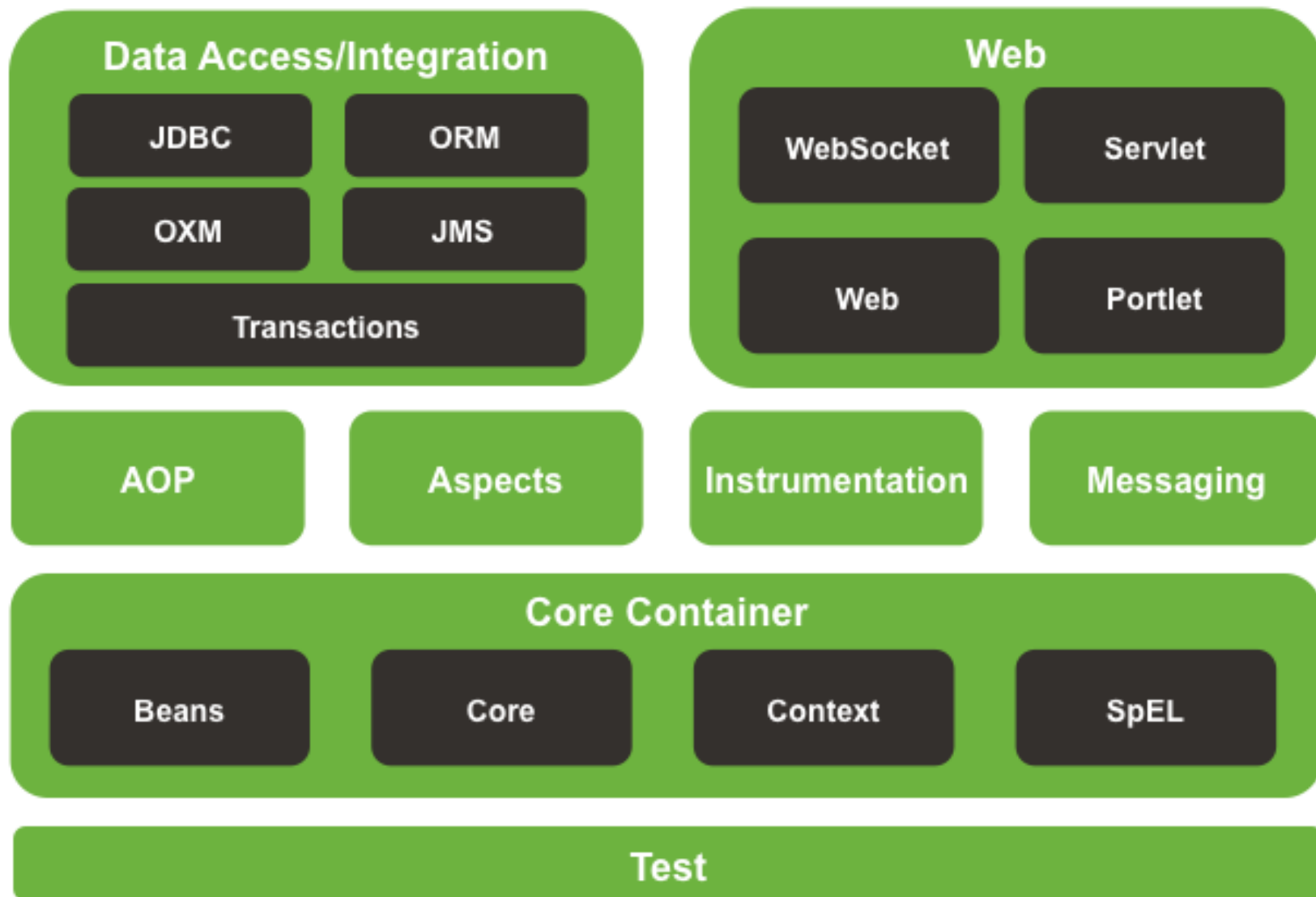
---

- Framework open-source criado por Rod Johnson
- Criado para simplificar o desenvolvimento empresarial
- Inversão de controle
- Injeção de dependência





# Spring Framework Runtime





# Spring

---

- Excesso de arquivos de configuração
- Dependência de bibliotecas (52 para versão web)



# Spring

```
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" version="3.1">

  <display-name>Spring MVC XML Configuration Example</display-name>

  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:app-config.xml</param-value>
  </context-param>

  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>

  <servlet>
    <servlet-name>my-dispatcher-servlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>classpath:web-config.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>my-dispatcher-servlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

</web-app>
```

# Spring Boot

---

- **Simplificar o início do desenvolvimento de aplicações Spring**
- **Elimina todos os xmls e configurações**
- **Configura todas as bibliotecas necessárias**
- **Disponibiliza servidor tomcat ou jetty embutido**



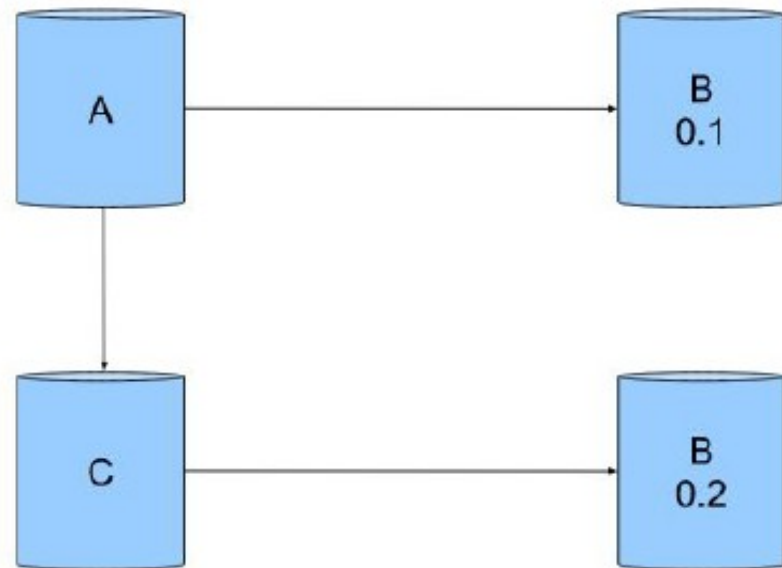
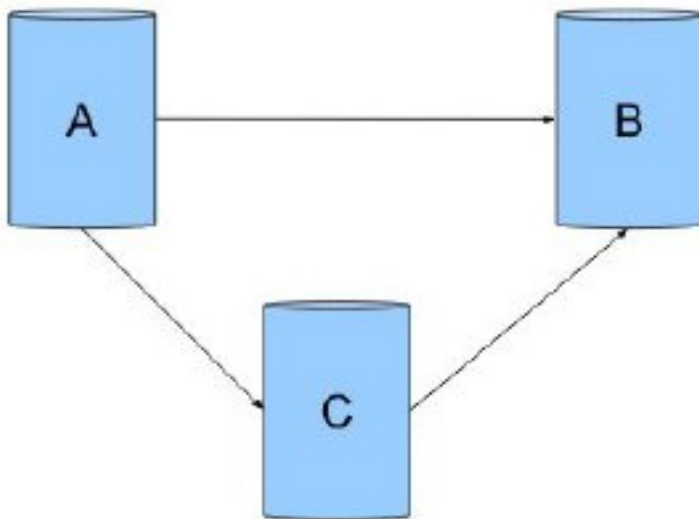
# Maven

---

- **Ferramenta para gerenciamento, construção e implantação de projetos**
- **Simplifica o processo de Build**
- **Organiza e compartilha as bibliotecas dos projetos**
- **Gera relatórios e documentação sobre o projeto**



# Maven



# Maven

---

- Builds automatizados
- Ciclo de vida bem definido (lifecycle)
- Repositório de artefatos
- Gestão de dependências
- Definição declarativa de projetos (POM)
- Plugins
- Início rápido de projetos
- Site de projetos facilmente acessível
- Integração com controle de versão



# Maven

---

- **Project Object Model (POM)**
  - **Arquivo xml**
  - **Definições do projeto**
  - **Configurações de dependências de componentes(lib)**
  - **Customizações (definições de plugin, repositórios, etc..)**
  - **Herança e agregação de projetos**



# Maven – pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
</project>
```





# Maven – pom.xml

Versão  
do Maven

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
</project>
```



# Maven – pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
</project>
```

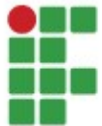
Nome  
do pacote



# Maven – pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
</project>
```

Identificador  
do projeto



# Maven – pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
</project>
```

Versão do projeto



# Maven – pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
</project>
```

Forma de  
Empacotamento  
(pom, war, jar, ...)



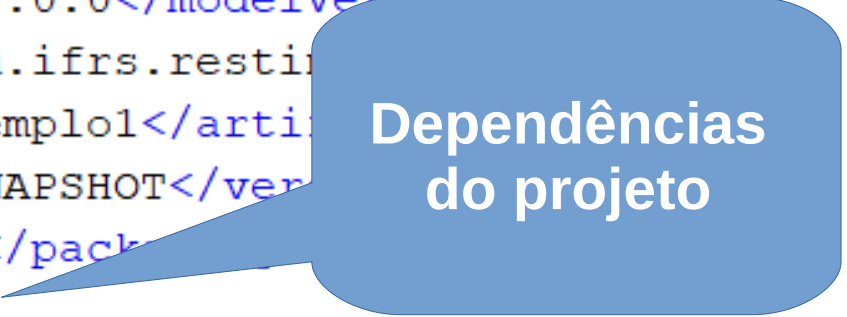
# Maven – Spring - pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.4.RELEASE</version>
  </parent>
</project>
```



# Maven – Spring - pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restitui</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.4.RELEASE</version>
  </parent>
</project>
```



Dependências  
do projeto

# Maven – Spring - pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-web</artifactId>
    </dependency>
  </dependencies>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.4.RELEASE</version>
  </parent>
</project>
```

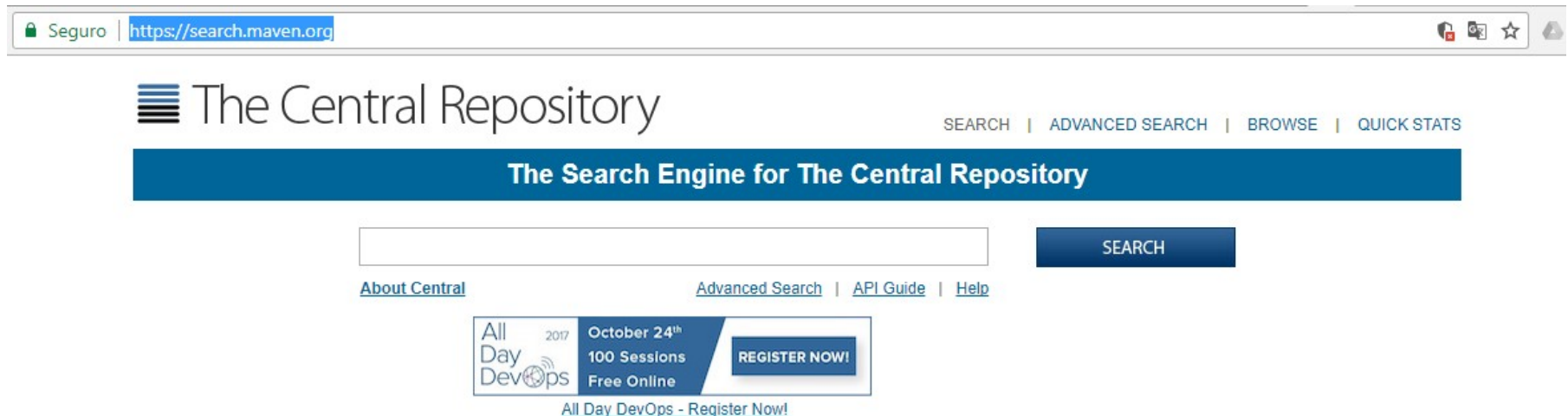


Herança do projeto

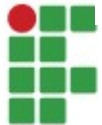
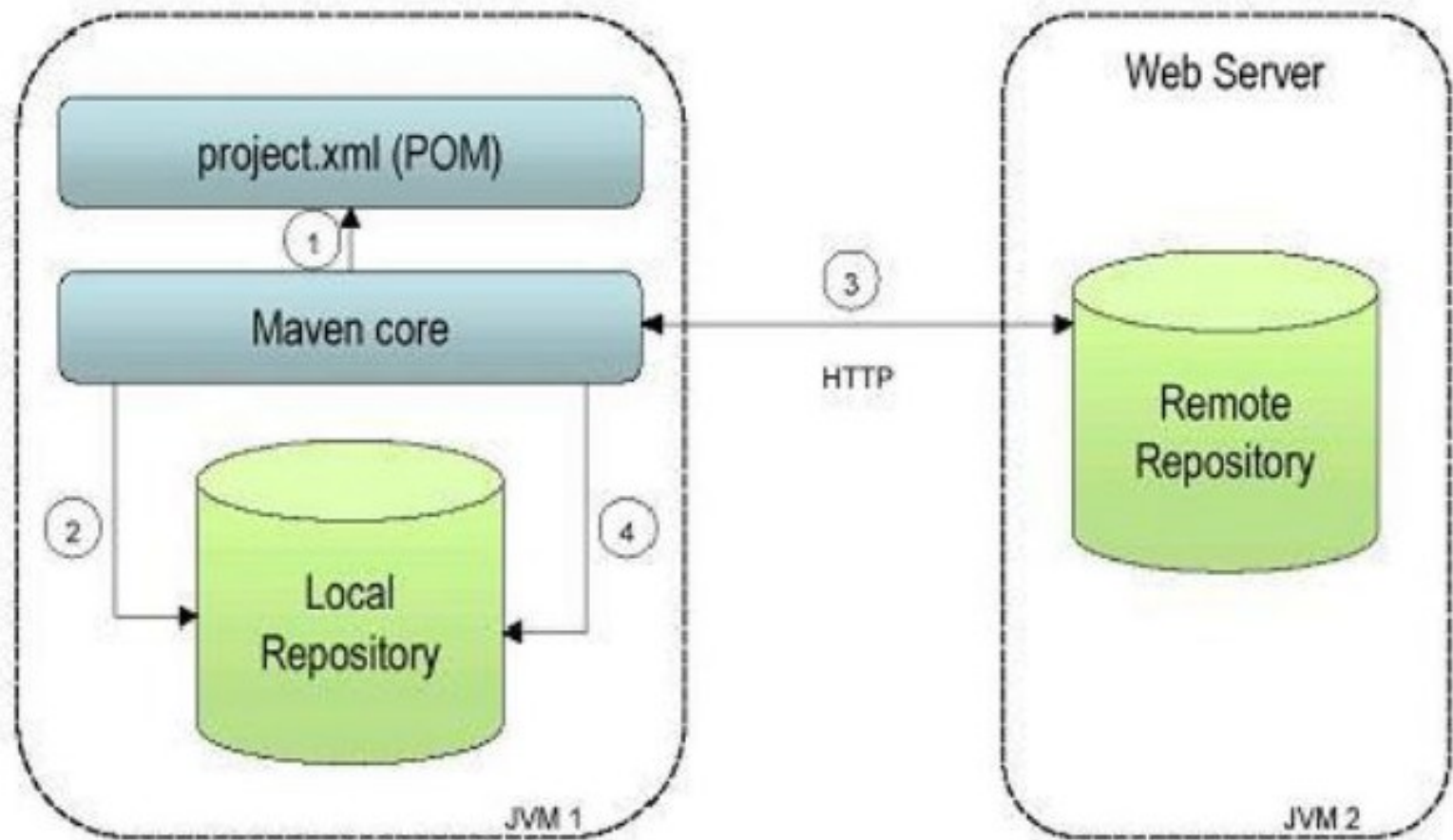


# Central de repositório

- <https://search.maven.org/>



# Central de repositório



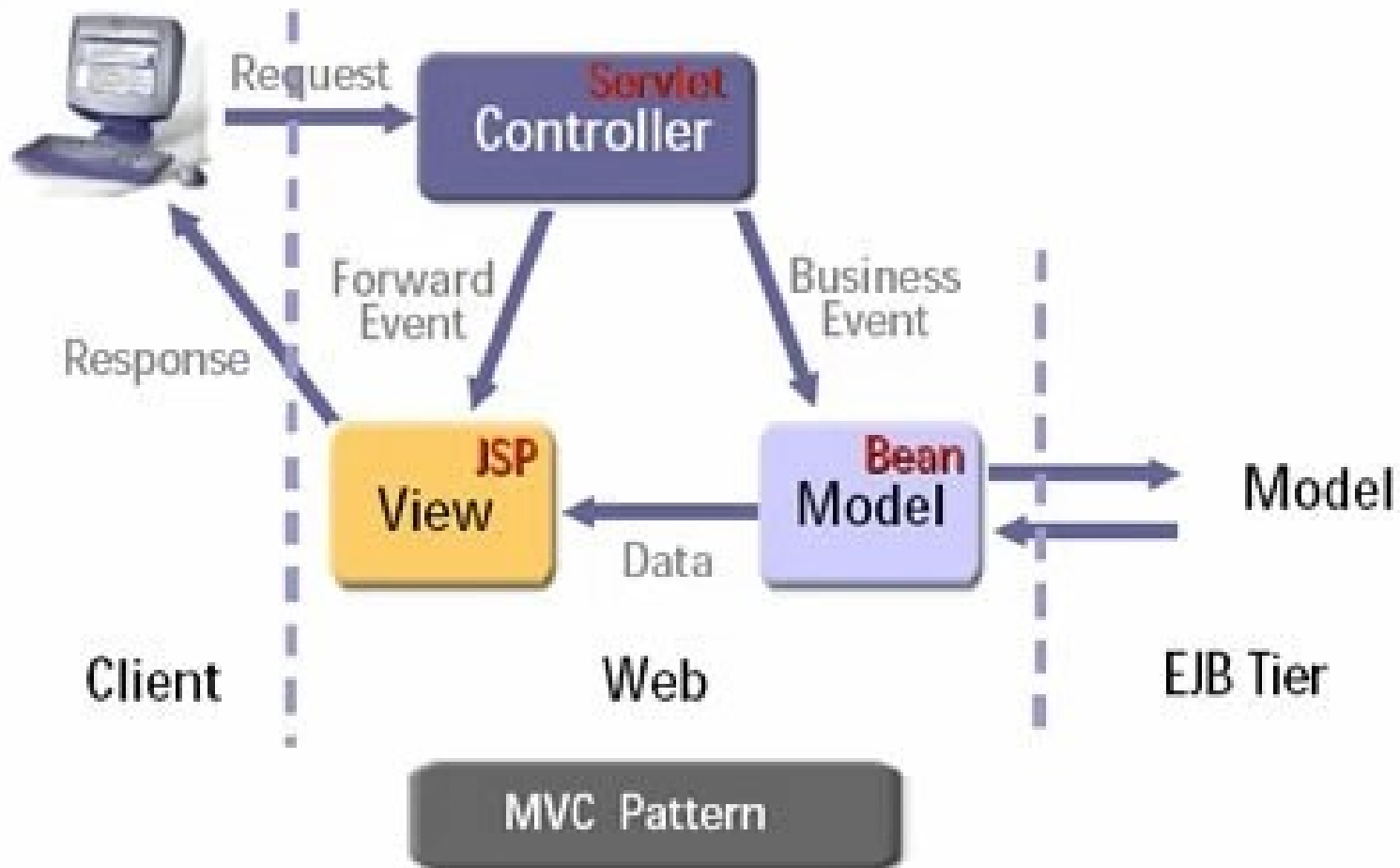
# MVC

---

- **Model View Controller**
- **Separa a lógica da aplicação (modelo) e a sua visualização (view)**
- **Modelo em três camadas**
  - **View = VISÃO**
  - **Model = MODELO**
  - **Controller = CONTROLADOR**



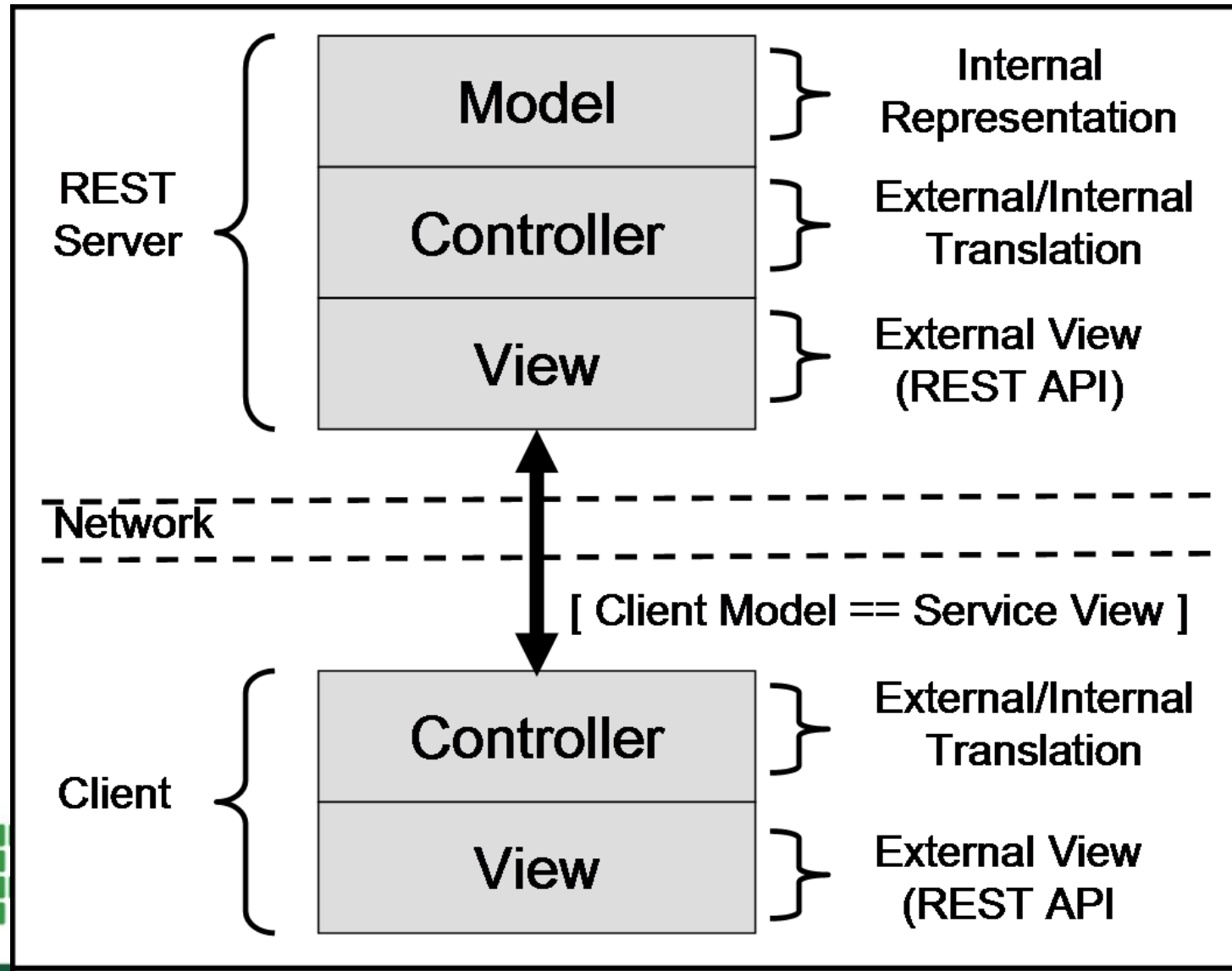
# JSP



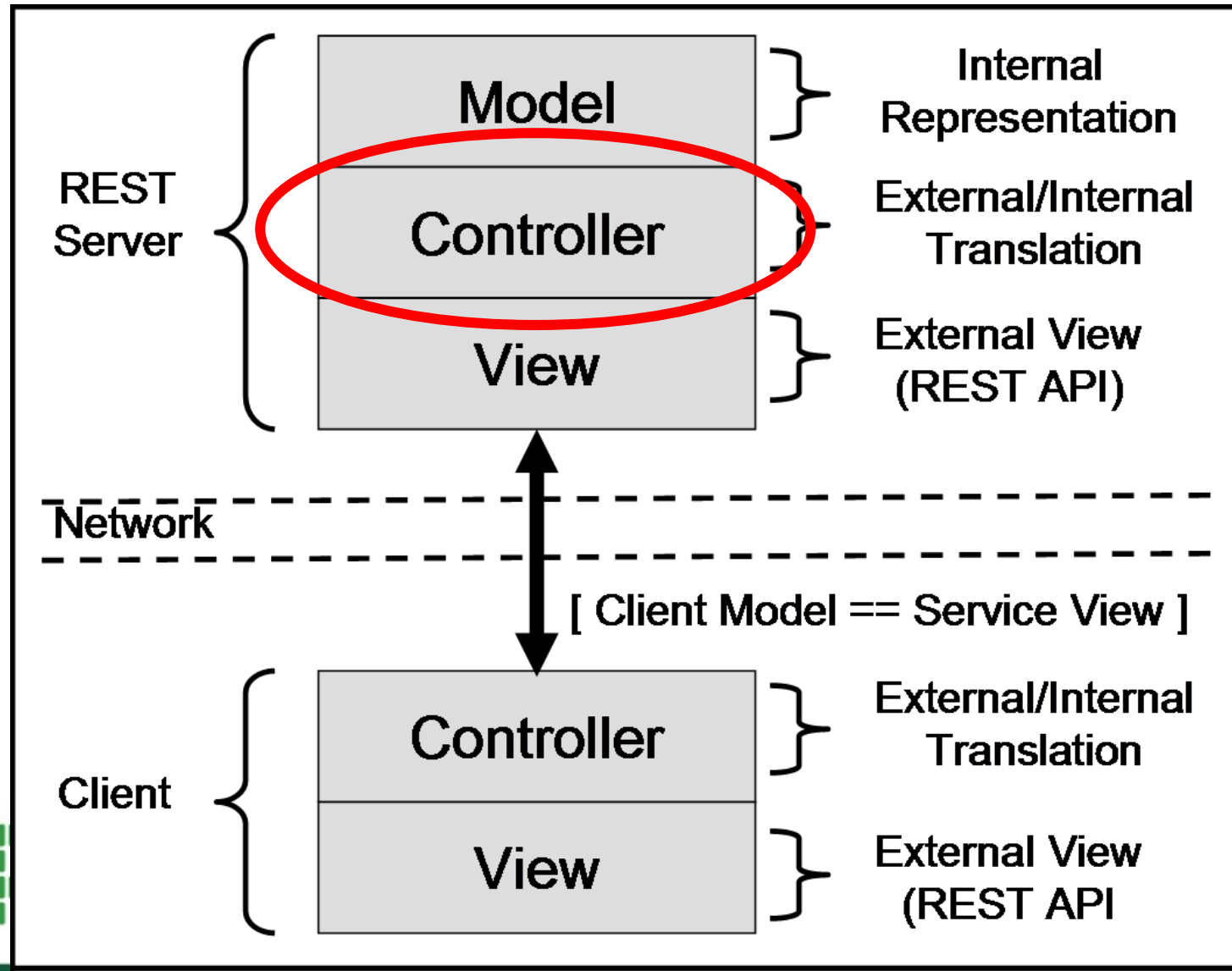
# JSP

```
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10  <head>
11      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12      <title>JSP Page</title>
13  </head>
14  <body>
15      <%! int result; %>
16
17      <jsp:useBean id="calcul" scope="page" class="Beans.Calculator" />
18
19      <jsp:setProperty name="calcul" property="lhs" param="lhs" />
20      <jsp:setProperty name="calcul" property="rhs" param="rhs" />
21      <jsp:setProperty name="calcul" property="oper" param="oper" />
22
23      <h1>Resultado:</h1>
24      <h2><%= calcul.getResult() %></h2>
25      <br/>
26      <a href="index.jsp">&DoubleLeftArrow; (back)</a>
27  </body>
28 </html>
```

# Spring + Rest



# Spring + Rest



# Maven – Spring - pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.edu.ifrs.restinga.dsw</groupId>
  <artifactId>Exemplo1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.4.RELEASE</version>
  </parent>
</project>
```





# Ferramenta de inicialização do Spring

---

- <https://start.spring.io/>



# Projeto Spring

---

```
package br.edu.ifrs.restinga.exemplo1;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Exemplo1 {

    public static void main(String[] args) {
        SpringApplication.run(Exemplo1.class, args);
    }
}
```



# Projeto Spring

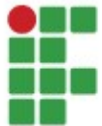
```
package br.edu.ifrs.restinga.exemplo1;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Exemplo1 {

    public static void main(String[] args) {
        SpringApplication.run(Exemplo1.class, args);
    }
}
```

Define que a aplicação é autoconfigurada



# Projeto Spring

---

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class ControllerExemplo {
    @RequestMapping("/")
    @ResponseBody
    public String ola() {
        return "Olá, mundo! ";
    }
}
```



# Projeto Spring

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class ControllerExemplo {
    @RequestMapping("/")
    @ResponseBody
    public String ola() {
        return "Olá, mundo! ";
    }
}
```

Define que a classe tem o comportamento de um controller

# Projeto Spring

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class ControllerExemplo {
    @RequestMapping("/")
    @ResponseBody
    public String ola() {
        return "Olá, mundo! ";
    }
}
```

Mapeia uma url em um método da classe controller  
"/" representa a raiz da aplicação

# Projeto Spring

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class ControllerExemplo {
    @RequestMapping("/")
    @ResponseBody
    public String ola() {
        return "Olá, mundo! ";
    }
}
```

Altera o comportamento para retornar uma string, ao invés de uma view padrão do spring

# Projeto Spring

---



Olá, mundo!





# Parâmetros

---

- **Três formas de enviar parâmetros para os métodos**
  - **Pelo próprio caminho da url (path), através do `@PathVariable`**
    - ♦ **127.0.0.1:8080/ola/jezer**
  - **Pelas variáveis das urls http, através do `@RequestParam`**
    - ♦ **127.0.0.1:8080/ola?nome=jezer**
  - **Pelo corpo da mensagem (POST)**



# Parâmetros

---

```
@Controller
public class ControllerExemplo {
    @RequestMapping("/")
    @ResponseBody
    public String ola() {
        return "Olá, mundo! ";
    }

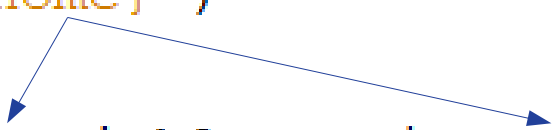
    @RequestMapping("/ola/{nome}")
    @ResponseBody
    public String ola(@PathVariable String nome) {
        return "Olá, "+nome+" !";
    }
}
```



# Parâmetros

```
@Controller
public class ControllerExemplo {
    @RequestMapping("/")
    @ResponseBody
    public String ola() {
        return "Olá, mundo! ";
    }

    @RequestMapping("/ola/{nome}")
    @ResponseBody
    public String ola(@PathVariable String nome) {
        return "Olá, "+nome+" !";
    }
}
```



# Parâmetros

---



Olá, jezer !

# Parâmetros

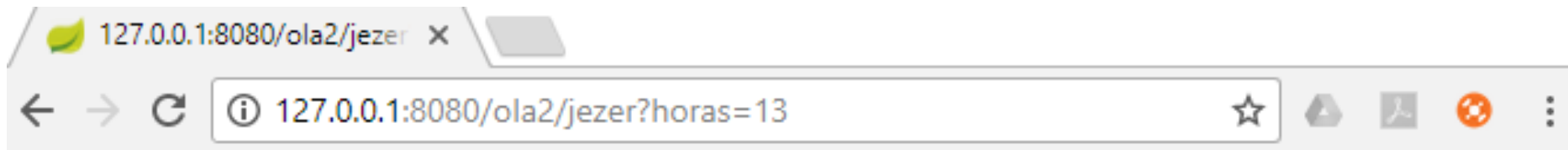
---

```
@RequestMapping("/ola2/{nome}")
@ResponseBody
public String ola(@PathVariable String nome,
                  @RequestParam("horas") int horas){
    if(horas>18)
        return "Boa noite, "+nome+" !";
    if(horas>12)
        return "Boa tarde, "+nome+" !";
    else
        return "Boa dia, "+nome+" !";
}
```



# Parâmetros

---

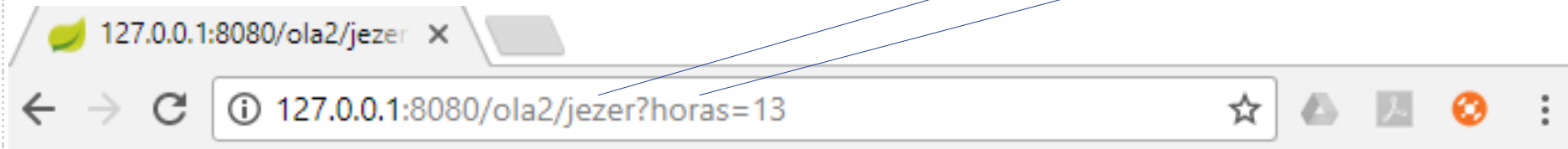


Boa tarde, jezer !



# Parâmetros

```
@RequestMapping("/ola2/{nome}")  
@ResponseBody  
public String ola(@PathVariable String nome,  
                  @RequestParam("horas") int horas) {
```



Boa tarde, jezer !