

INSTITUTO FEDERAL
SÃO PAULO



Introdução ao Spring e Gerenciador de pacotes Maven

Desenvolvimento de Componentes (BRADECO)

Prof. Luiz Gustavo Diniz de Oliveira Vêras

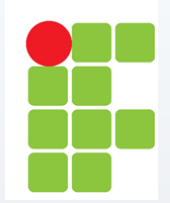
E-mail: gustavo_veras@ifsp.edu.br



Roteiro da aula

- ✓ **Spring Framework**
- ✓ **Criando um projeto do Spring no VS Code**
- ✓ **Gerenciador de pacotes Maven**
- ✓ **Conceitos básicos do Spring**

Spring Framework



- O Spring Framework fornece um modelo abrangente de programação e configuração para aplicativos empresariais modernos (especialmente projetos para a Web) baseados em Java - em qualquer tipo de plataforma de implantação.
- **Um elemento-chave do Spring é o suporte de infraestrutura no nível do aplicativo:**
 - Assim as equipes de desenvolvimento podem se concentrar na lógica de negócios;
 - Evita esforços desnecessários com ambientes de implementação específicos na fase de desenvolvimento do aplicativo (programação);



Spring Framework

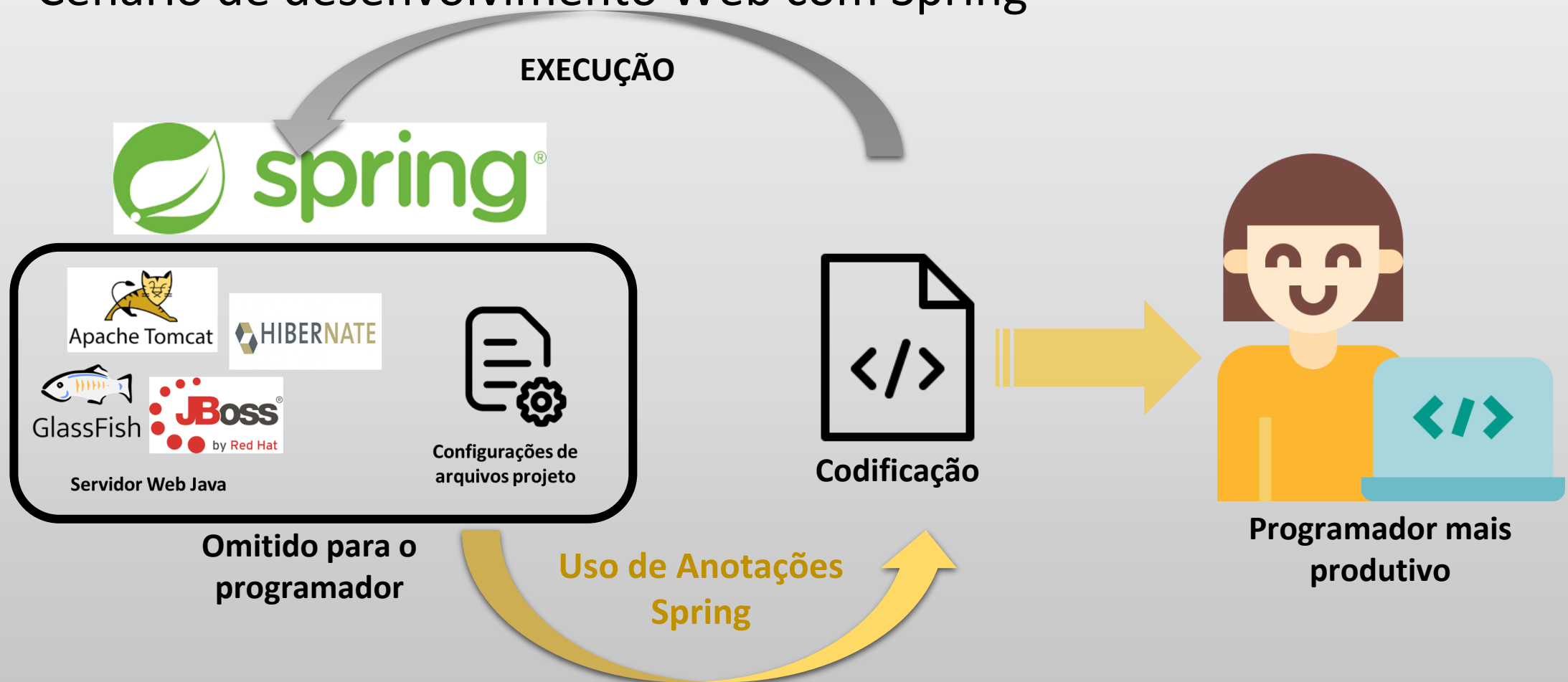
- Cenário tradicional de desenvolvimento Web com Java





Spring Framework

- Cenário de desenvolvimento Web com Spring





Família Spring

Alguns produtos da família Spring

- **Spring Boot:** facilita a criação de aplicativos autônomos baseados em Spring de nível de produção que você pode "simplesmente executar".
- **Spring Security:** Spring Security é uma estrutura de autenticação e controle de acesso poderosa e altamente personalizável. É o padrão de fato para proteger aplicativos baseados em Spring. Vamos utiliza-lo para configurar uma funcionalidade de Login.
- **Spring Data JPA:** visa melhorar significativamente a implementação de camadas de acesso a dados (Model), reduzindo o esforço para a quantidade que é realmente necessária de código para realizar persistência em banco de dados.



Família Spring

Alguns produtos

- **Spring Boot** Spring de
- **Spring Security** controle fato para configura
- **Spring Data** camadas quantidade persistência em banco de dados.

Veja a quantidade de produtos oferecidos pela Spring

<https://spring.io/projects/spring-framework>

dados em ar".

icação e padrão de a-lo para

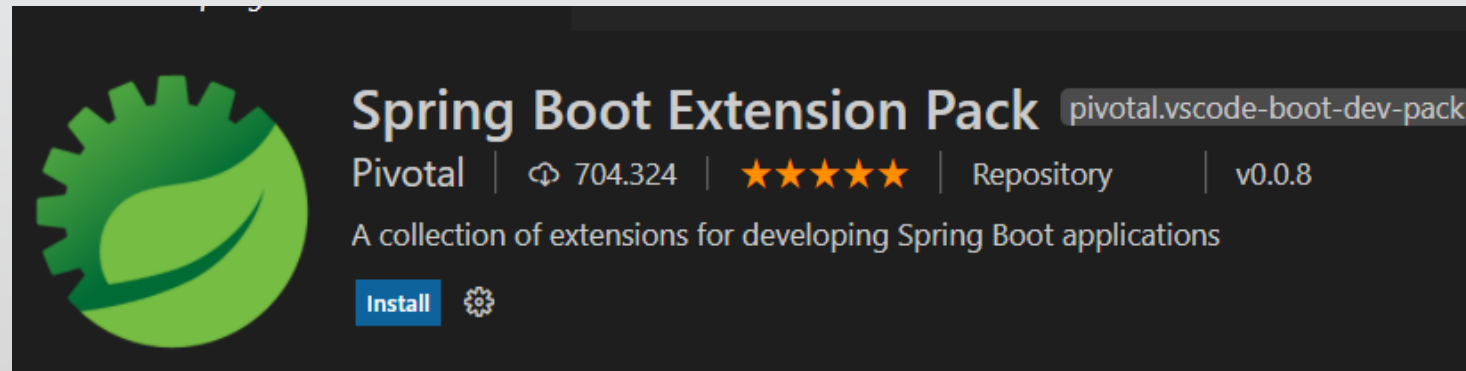
tação de para a realizar

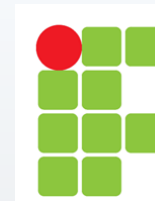


Trabalhando com Spring no VS Code

Primeiro, devemos instalar o plugin para trabalhar com o Spring

- Spring boot Extension pack



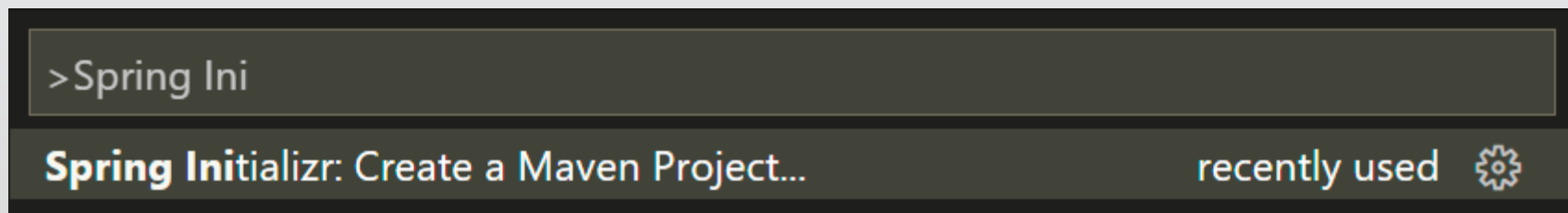


Trabalhando com Spring no VS Code

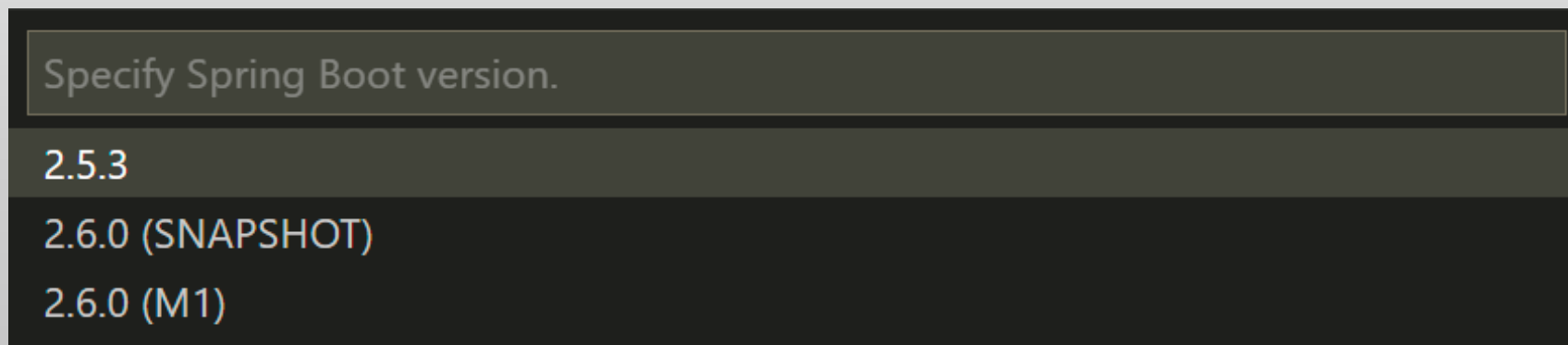
Criando o projeto

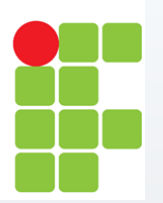
No VS Code, pressione CTRL + SHIFT + P.

- Digite *Spring Initializr* e Selecione a opção que contém *Maven Project*.



- Selecione a versão mais recente (que não tenha escrito SNAPSHOT)





Trabalhando com Spring no VS Code

Criando o projeto

- Selecione *Java* como linguagem para o projeto.

Specify project language.

Java

- O Spring solicita que você defina um id de grupo (algo que faça referência ao domínio da sua aplicação) para o projeto. Digite um nome com o padrão de nomeação de pacote.

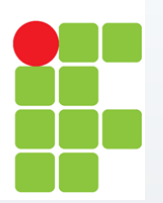
com.ifsp

Input Group Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)

- O Spring solicita que você defina um id de artefato. Esse é o nome do projeto que irá trabalhar a partir daqui.

exemplo

Input Artifact Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)



Trabalhando com Spring no VS Code

Criando o projeto

- Selecione *Jar* o tipo de empacotamento.

Specify packaging type.

Jar

- Selecione a versão do Java que está instalado na sua máquina. Aqui usaremos a versão 11.

Specify Java version.

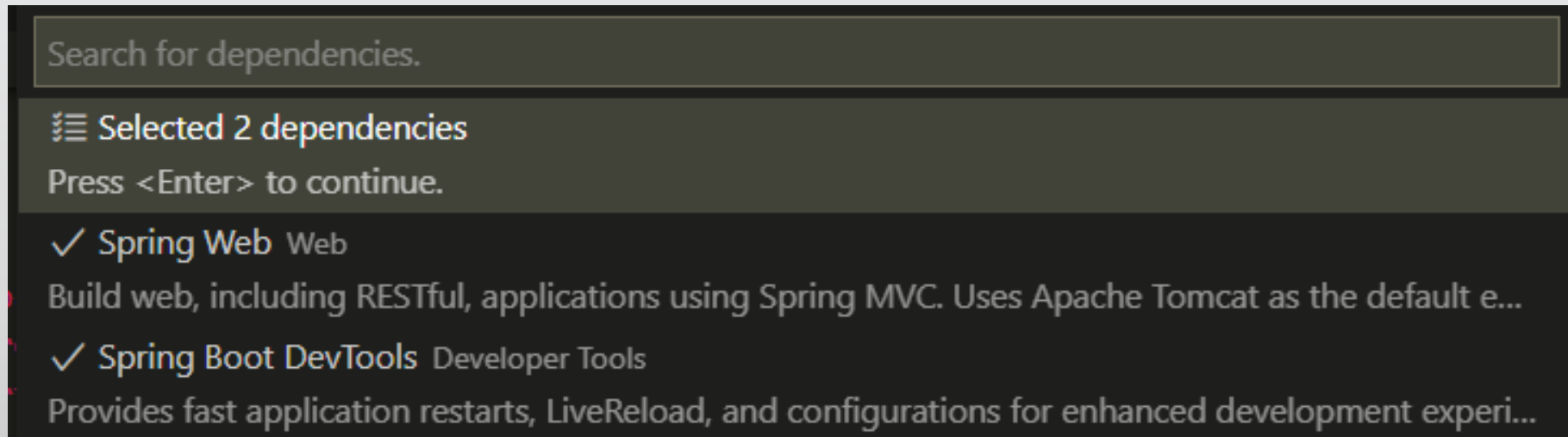
11



Trabalhando com Spring no VS Code

Criando o projeto

- Selecione as dependências (bibliotecas) que iremos utilizar.



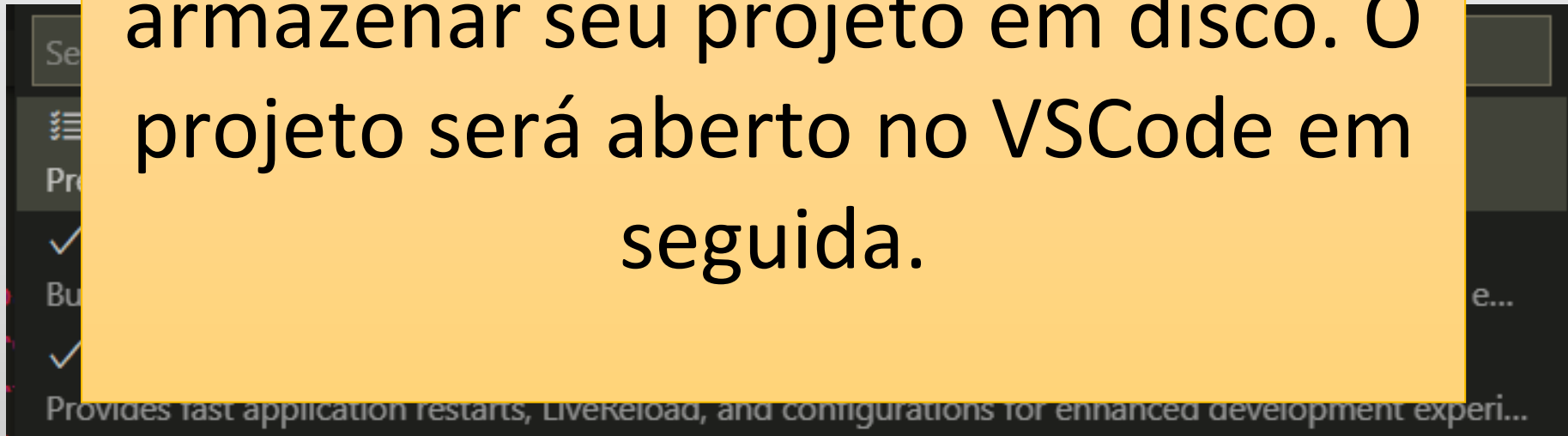


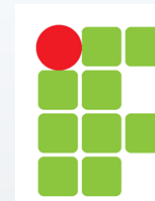
Trabalhando com Spring no VS Code

Criando o

- Selecion

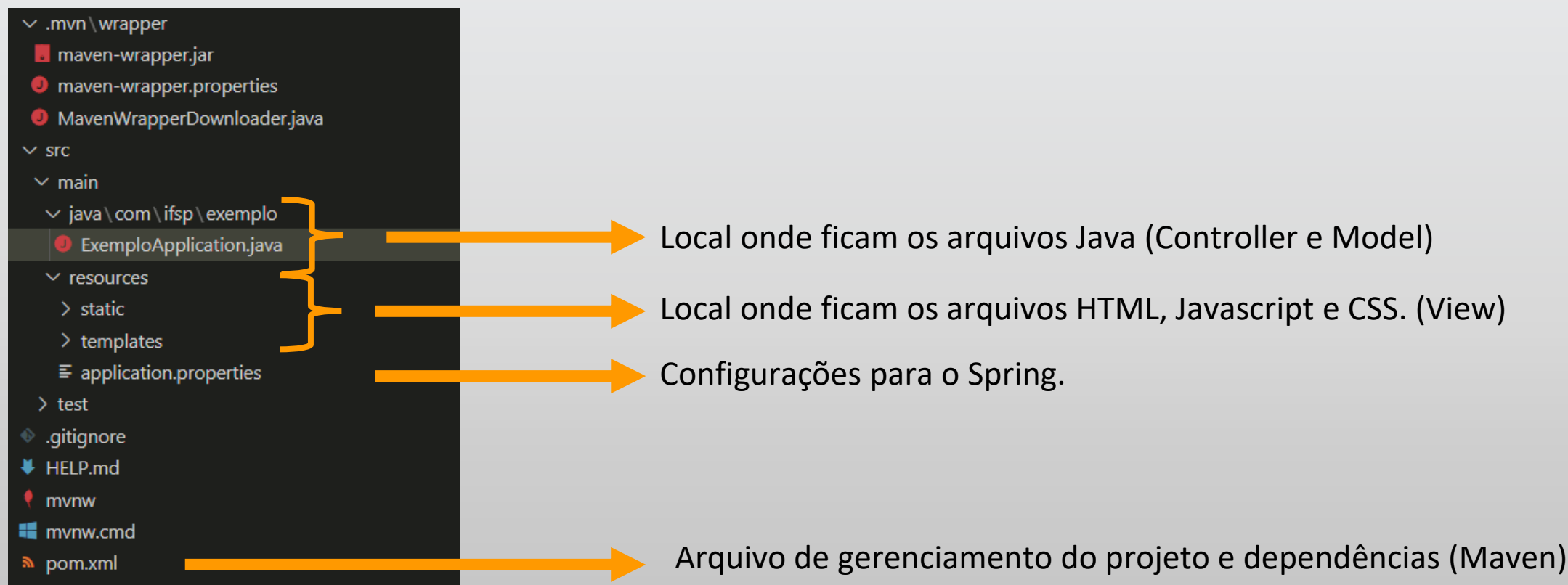
Por fim, selecione um local para armazenar seu projeto em disco. O projeto será aberto no VSCode em seguida.





Trabalhando com Spring no VS Code

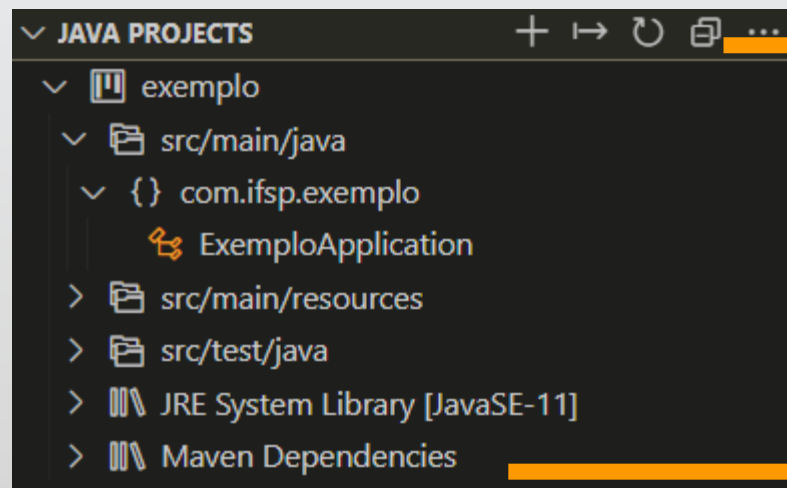
Após selecionar o diretório onde o projeto será gerado, teremos a estrutura a seguir:





Trabalhando com Spring no VS Code

Outras abas que podem ser úteis durante o desenvolvimento



Permite configurar o projeto da percepção de projeto Java

Dependências da aplicação criadas pelo Maven



Trabalhando com Spring no VS Code

Abra o arquivo **ExemploApplication.java** e deixe como segue

```
@SpringBootApplication
@RestController
public class DemoApplication {

    @GetMapping("/hello")
    String home(){
        return "Hello World!";
    }

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




- ```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text, lexclude) Spring Boot-ExemploApp ⌵ ≡ ^ ×
```

```
. _ . _ _ _ _ \\\ / ____'_____()_ __ _ _ \\ \\ \\ \\
(()__ | '_| |_||'_V_'_| \\ \\ \\ \\
\\W ___)| |_| ||| ||| ||(| |)))
' |___| .__||_| | |_,| // //
=====|_|=====|_/=/// //
:: Spring Boot :: (v2.5.3)
```

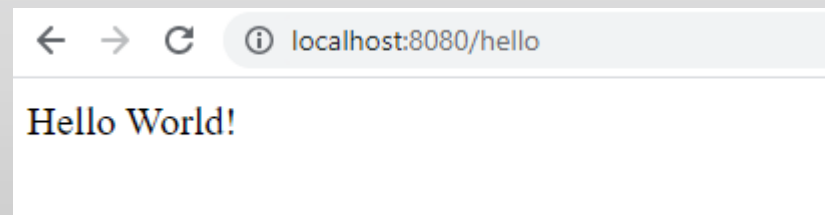
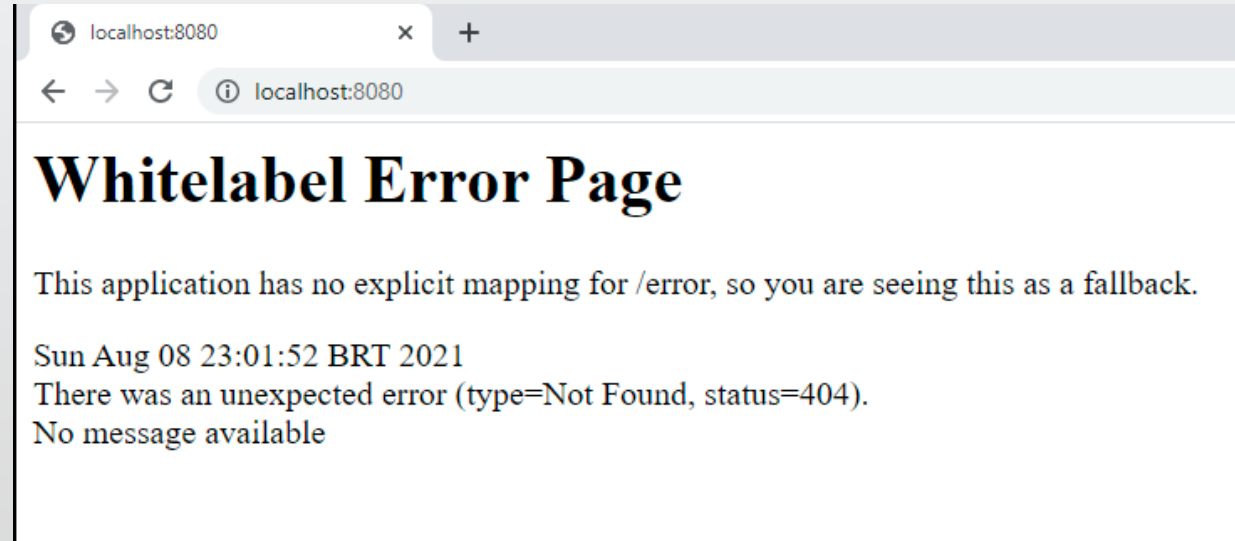
```
2021-08-08 23:01:15.726 INFO 11268 --- [restartedMain] com.ifsp.exemplo.ExemploApplication : Starting ExemploApplication using Java 15 on DESKTOP
-LOVKP02 with PID 11268 (C:\Java Projects (VS)\SpringBoot\exemplo\target\classes started by luiz_gustavo in c:\Java Projects (VS)\SpringBoot\exemplo)
2021-08-08 23:01:15.729 INFO 11268 --- [restartedMain] com.ifsp.exemplo.ExemploApplication : No active profile set, falling back to default profi
```

- ```
2021-08-09 11:36:53.702 INFO 14824 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
```



Trabalhando com Spring no VS Code

- Acesse no browser o endereço **localhost:8080** e veja o que aparece.
- Agora tente digitar **/hello** na frente do endereço e tente acessar.





Gerenciador de dependências Maven



- O Maven é uma ferramenta de gerenciamento e automação de construção (build) de projetos e dependências.

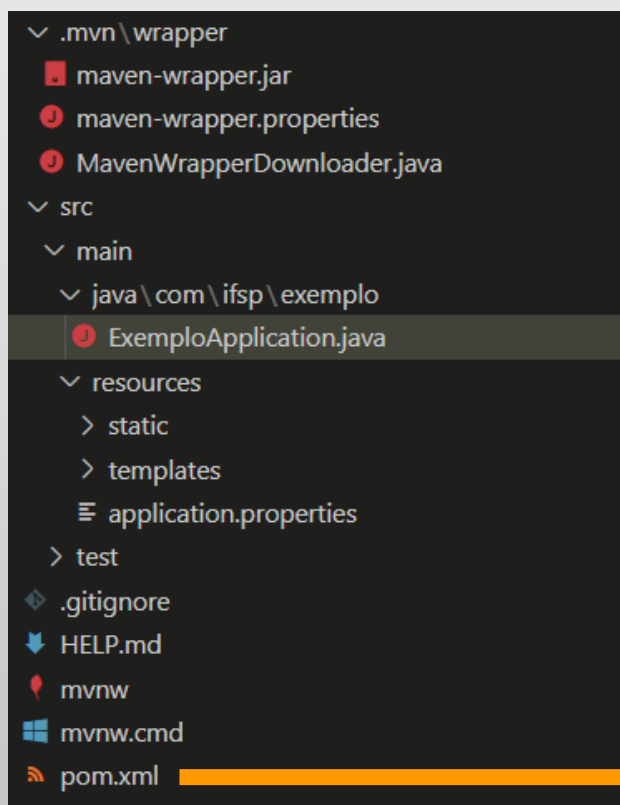
Vantagens:

- Um programador novo que tem contato com um projeto não precisa de imediato saber quais dependências (bibliotecas) o projeto necessita para compilar e executar,
- Ele não precisará descobrir onde obtê-las e nem irá se preocupar em como realizar a construção do pacote do aplicativo.
- Facilita o compartilhamento de projetos.
- O Maven automatiza as tarefas de importação das dependências/bibliotecas (.jar) e já as adiciona para o programador no classpath.



Gerenciador de dependências Maven

- O Spring já configura automaticamente o Maven na criação do projeto.
- Podemos adicionar novas dependências alterando o arquivo **pom.xml**



Arquivo de gerenciamento do projeto e dependências (Maven)

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>  <!-- VERSÃO DO POM CONSIDERADA -->
  <parent> <!-- ESPECIFICA QUE O NOSSO PROJETO É DO TIPO SPRING. -->
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.5.3</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <!-- INFORMAÇÕES SOBRE O NOSSO PROJETO. -->
  <groupId>com.ifsp</groupId>
  <artifactId>exemplo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>exemplo</name>
  <description>Demo project for Spring Boot</description>
  <!-- PROPRIEDADES E CONFIGURAÇÕES DO PROJETO. -->
  <properties>
    <java.version>11</java.version>
  </properties>
  <dependencies>
    <!-- ONDE INCLUIMOS AS DEPÊNDENCIAS. PARTE QUE MAIS IREMOS ALTERAR -->
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```



Gerenciador de dependências Maven

Olhando melhor quais dependências estão no nosso projeto.

pom.xml

```
<!-- TRECHO RESTANTE OMITIDO [?]
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
<!-- TRECHO RESTANTE OMITIDO [?]
```



Gerenciador de dependências Maven

Olhando melhor quais dependências estão no nosso projeto.

pom.xml

```
<!-- TRECHO RESTANTE OMITIDO [?]
```

```
<dependencies>
```

```
  <dependency>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-web</artifactId>
```

```
  </dependency>
```

Importa a biblioteca do Spring para trabalhar com web e podemos criar a API.

```
  <dependency>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-devtools</artifactId>
```

```
    <scope>runtime</scope>
```

```
    <optional>>true</optional>
```

```
  </dependency>
```

Utilizado quando executamos a aplicação “em produção”

```
  <dependency>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-test</artifactId>
```

```
    <scope>test</scope>
```

```
  </dependency>
```

Utilizado só quando estamos realizando testes com a aplicação (não coberto neste curso).

```
</dependencies>
```

```
<!-- TRECHO RESTANTE OMITIDO [?]
```



Gerenciador de dependências Maven

Olhando m

pom.xml

```
<!-- TRECHO RESTANTE  
<dependencies>  
  <dependency>  
    <groupId>  
    <artifactId>  
  </dependency>  
  
  <dependency>  
    <groupId>  
    <artifactId>  
    <scope>ru  
    <optional>  
  </dependency>  
  <dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-test</artifactId>  
    <scope>test</scope>  
  </dependency>  
</dependencies>  
<!-- TRECHO RESTANTE OMITIDO
```

Como descobrir como inserir novas dependências/bibliotecas?

<https://mvnrepository.com/>

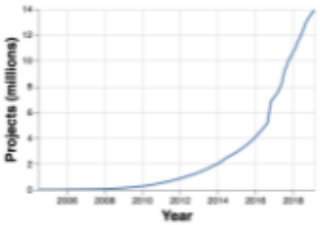
Utilizado só quando estamos realizando testes com a aplicação (não coberto neste curso).

Maven Repository: Search/Browse x +

mvnrepository.com

MVNREPOSITORY Search for groups, artifacts, categories Search Categories | Popular | Contact Us

Indexed Artifacts (23.0M)



Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases
- HTML Parsers
- HTTP Clients

What's New in Maven

Docker Test Utilities For OfficeFloor 5 usages
net.officefloor.docker » officedocker_test » 3.39.0 AGPL
Provides utilities for working with Docker for third-party services
Last Release on Aug 8, 2021

OfficeFloor Filing Cabinet 4 usages
net.officefloor.cabinet » officecabinet » 3.39.0 AGPL
Cloud storage for OfficeFloor
Last Release on Aug 8, 2021

Polyglot Test 4 usages
net.officefloor.polyglot » officepolyglot_test » 3.39.0 AGPL
Tests for building Polyglot plugins
Last Release on Aug 8, 2021

OfficeCompiler 89 usages
net.officefloor.core » officecompiler » 3.39.0 AGPL
Compiler for OfficeFloor (compiles the sources into the frame for execution)
Last Release on Aug 8, 2021

Indexed Repositories (1337)

- Central
- Sonatype
- Spring Plugins
- Atlassian
- Spring Lib M
- Hortonworks
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

Popular Tags

android apache ant annotation

Busque pela dependência aqui.
Por exemplo, MySQL





Indexed Artifacts (23.0M)

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases
- HTML Parsers
- HTTP Clients

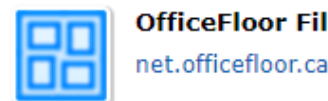
What's New in Maven



Docker Test UI
net.officefloor.doc
officefloor_test

Provides utilities
third-party serv

Last Release on A



OfficeFloor File
net.officefloor.ca

Cloud storage fo

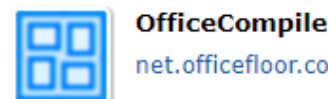
Last Release on A



Polyglot Test
net.officefloor.po
3.39.0

Tests for buildin

Last Release on A



OfficeCompiler
net.officefloor.co

Compiler for Off
sources into the

Last Release on A

Maven Repository: mysql » mysql x +

mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.26

MVNREPOSITORY

Search for groups, artifacts, categories

Home » mysql » mysql-connector-java » 8.0.26

MySQL Connector/J » 8.0.26
JDBC Type 4 driver for MySQL

License	GPL 2.0
Categories	MySQL Drivers
Organization	Oracle Corporation
HomePage	http://dev.mysql.com/doc/connector-j/en/
Date	(Jul 19, 2021)
Files	pom (2 KB) jar (2.3 MB) View All
Repositories	Central
Used By	5,479 artifacts

[Maven](#) [Gradle](#) [Gradle \(Script\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#)

[Leiningen](#) [Buildr](#)

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.26</version>
</dependency>
```

Basta copiar este código e colar dentro da tag <dependencies> do arquivo pom.xml.



Conceitos de Spring Framework

O Spring está baseado em dois princípios de design:

- **Inversão de Controle (IoC - Inversion of Control)**

Neste princípio, a responsabilidade de controlar (criar, ativar, destruir) um objeto no sistema é transferida do programador para um outro elemento. A vantagem dessa abordagem é a facilidade em modificar e testar componentes do sistemas.

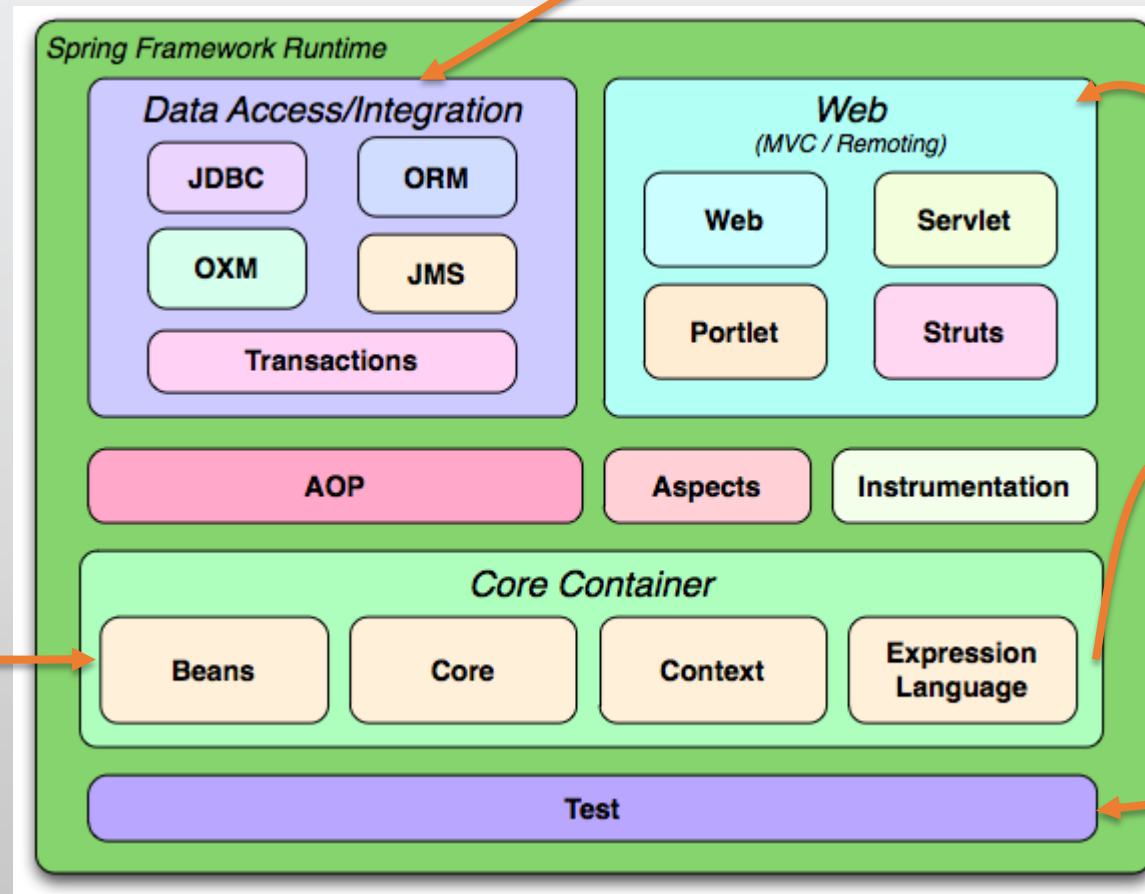
- **Injeção de Dependência (DI – Dependency Injection)**

Neste princípio, um elemento do sistema é responsável por carregar (resolver dependências) para o programador instâncias de um objeto na memória em uma parte específica de um sistema. É a forma como o Spring implementa a inversão de controle. Dentre suas vantagens temos a facilidade em substituir dependências sem alterar o código principal, código mais organizado e testável.



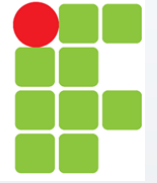
Conceitos de Spring Framework

Arquitetura do Spring Framework

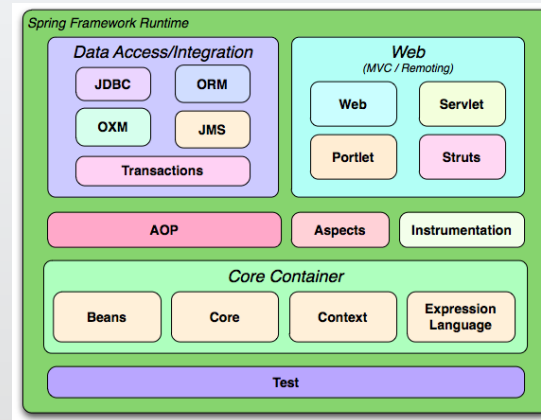


No Spring, um componente é criado em seu **container (Aplicação da Inversão de Controle)**...

... e carregado em diferentes partes da aplicação (**Aplicação da Injeção de Dependência**).



Conceitos de Spring Framework



- **Core Container:** Contém os módulos essenciais do Spring, incluindo:
 - **Beans:** Gerenciamento de beans (objetos gerenciados pelo Spring).
 - **Core:** Fornece funcionalidades básicas do framework (IoC e DI) .
 - **Context:** Suporte para injeção de dependências e acesso a beans.
- **Data Access/Integration:** Facilita a comunicação com bancos de dados e sistemas de mensagens.
- **Web (MVC/Remoting):** Suporte para desenvolvimento web.
- **AOP (Aspect-Oriented Programming):** Permite programação orientada a aspectos (configuração de classes de logs, segurança e onde serão aplicadas pelo Spring).
- **Test:** Suporte para testes unitários e de integração no Spring.



Conceitos de Spring Framework

Exemplo

Em uma aplicação sem IoC o programador é responsável por instanciar cada objeto necessário para uma Tarefa...

```
public class Programa {  
  
    private Tarefa tarefa;  
  
    public void do() {  
        this.tarefa = new Tarefa();  
        tarefa.executar();  
    }  
}
```

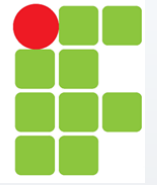


Conceitos de Spring Framework

Exemplo

...e também por determinar quando uma Tarefa deve ser executada.

```
public class Programa {  
    private Tarefa tarefa;  
  
    public void do() {  
        this.tarefa = new Tarefa();  
        tarefa.executar();  
    }  
}
```

Conceitos de Spring Framework

Exemplo

Na Inversão de Controle
podemos delegar a criação de
um objeto a outro elemento



No Spring chamamos esse elemento
de contêiner

E com isso uma classe pode se preocupar
menos em criar objetos e gerenciar suas
dependências e focar nas regras de negócio
que ela implementa

```
public class Programa {  
  
    private Tarefa tarefa;  
  
    public void do() {  
        tarefa.executar();  
    }  
}
```




Conceitos de Spring Framework

Exemplo

Na Inversão de Controle
podemos delegar a criação
um objeto a outro elemen

No Spring chamamos esse elemento
de contêiner

Como fazer isso no Spring?

le se preocupar
gerenciar suas
regras de negócio
enta

grama {
fa tarefa;

```
public void do() {  
    tarefa.executar();  
}  
}
```



Conceitos de Spring Framework

#1 – Defina uma interface

```
public interface Tarefa {  
    void executar();  
}
```

#2 – Crie uma classe que implemente a Tarefa e defina como componente com a anotação **@Component**.

```
import org.springframework.stereotype.Component;  
  
@Component  
public class TarefaImpl implements Tarefa {  
    @Override  
    public void executar() {  
        System.out.println("Executando tarefa...");  
    }  
}
```

#3 – Utilize em outro componente. O Spring injeta utilizando a anotação **@Autowired**.

```
@Component  
public class Programa {  
    private final Tarefa tarefa;  
  
    @Autowired  
    public Programa(Tarefa tarefa) {  
        this.tarefa = tarefa;  
    }  
  
    public void executar() {  
        tarefa.executar();  
    }  
}
```

Obs: Cada código deve ser criado em um arquivo .java separado.



Conceitos de Spring Framework

Altere o programa principal como segue.

```
@SpringBootApplication
public class Aplicacao implements CommandLineRunner {

    private final Programa programa;

    public Aplicacao(Programa programa) {
        this.programa = programa;
    }

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

    @Override
    public void run(String... args) {
        programa.executar();
    }
}
```

A saída “Executando tarefa...”
deve surgir no terminal.