

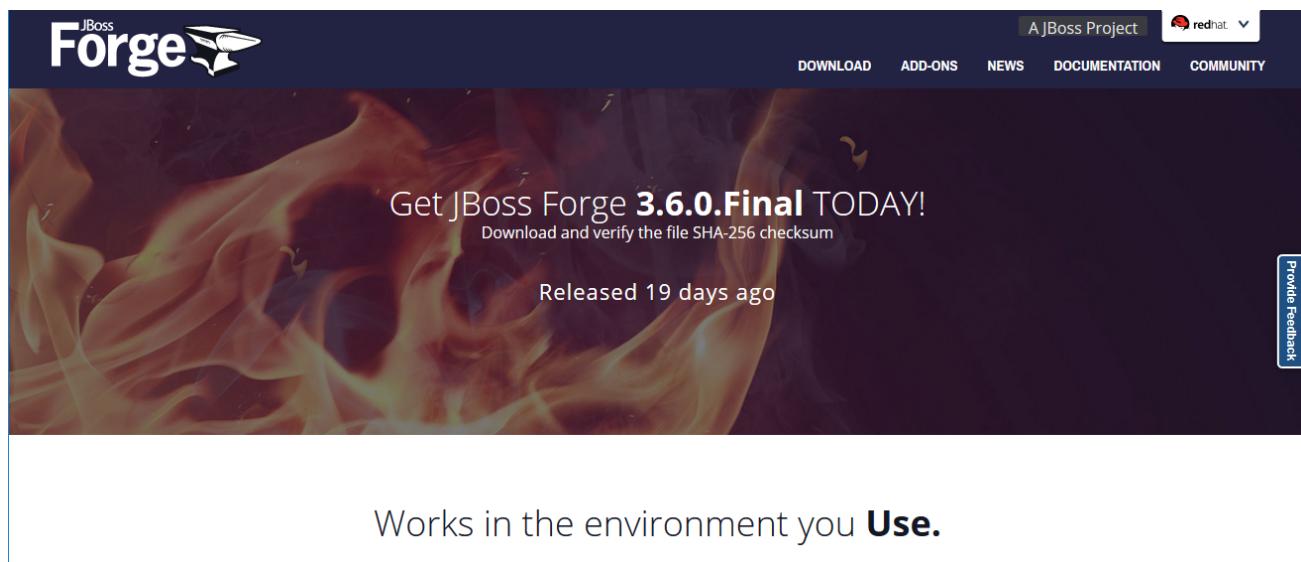
02

Criando o Projeto

Transcrição

Para nossos estudos, usaremos como base o projeto da [Casa do Código](https://www.casadocodigo.com.br/) (<https://www.casadocodigo.com.br/>). Além de ser um projeto bem conhecido, onde criaremos alguns cadastros, faremos o acompanhamento de livros, autores, preços e iremos mais a fundo com nosso projeto criando outras funcionalidades.

Para não perder tempo com a configuração inicial, faça o download do [JBoss Forge 3.0.1.Final](https://forge.jboss.org/download) (<https://forge.jboss.org/download>). Escolha seu sistema operacional, e faça o download para sua máquina.



Além disso, vamos rodar nosso projeto com o Application Server **Wildfly**. A versão que utilizaremos será a 10.0.0.Final. A versão mais atual pode ser outra, mas tenha cuidado ao pegar uma versão muito diferente, pois podemos ter problemas de configuração. Você pode baixar o Wildfly no endereço <http://wildfly.org/downloads/> (<http://wildfly.org/downloads/>).

Após realizar o download, descompacte os arquivos. Renomeia a pasta que você descompactou o JBossForge para apenas `forge`, assim será mais fácil o acesso via terminal.

Abra o seu **Terminal** e navegue até seu *workspace*. Dentro do seu workspace, aponte o terminal para o local onde você descompactou o forge. No meu caso:

```
./forge/bin/forge
```

Pressione `Enter` e assim já estaremos dentro do JBoss Forge. Depois, podemos digitar comando `forge` que criará o nosso projeto inicial. Assim digite no terminal

```
project-new --named casadocodigo
```

Os seus passos podem ser um pouco diferentes do realizado no vídeo, dependendo da versão do Forge utilizada, mas o resultado final será muito próximo.

Ao ser perguntado pelo pacote, digite `br.com.casadocodigo` .

Na *Version*, pode deixar a que foi sugerida, apenas pressione *ENTER*.

No *Final Name*, também pressione apenas *ENTER*.

No *Project Location*, apenas pressione *ENTER*.

No *Project Type*, se WAR já estiver selecionado, apenas pressione *ENTER*. Senão, digite a opção para WAR.

No *Build System*, se MAVEN já estiver selecionado, apenas pressione *ENTER*. Senão, digite a opção para MAVEN.

Na opção de *Stack*, vamos apenas digitar *ENTER* sem selecionar nada.

Agora você pode abrir a pasta do seu *Workspace* e navegar até o projeto criado `casadocodigo` . Você vai perceber que temos apenas uma pasta `src` e um arquivo `pom.xml` . Dentro da pasta `src` , temos apenas a sub estrutura de pacotes que informamos na configuração acima.

Agora que já temos a base, vamos configurar nossa stack. Voltando ao *Terminal*, digite

```
faces-setup --facesVersion 2.2
```

Logo depois, configuramos para usar o CDI, que será o 1.1.

```
cdi-setup --cdiVersion 1.1
```

Agora conseguimos alguma configuração no nosso `pom.xml` . Seu arquivo deve estar parecido com o de abaixo:

```
<?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-
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0
<modelVersion>4.0.0</modelVersion>
<groupId>br.com.casadocodigo</groupId>
<artifactId>casadocodigo</artifactId>
<version>1.0.0-SNAPSHOT</version>
<build>
  <finalName>casadocodigo</finalName>
  <plugins>
    <plugin>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.6</version>
      <configuration>
        <failOnMissingWebXml>false</failOnMissingWebXml>
      </configuration>
    </plugin>
  </plugins>
</build>
<packaging>war</packaging>
<properties>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
```

```
</properties>
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.jboss.spec</groupId>
      <artifactId>jboss-javaee-6.0</artifactId>
      <version>3.0.3.Final</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
    <dependency>
      <groupId>javax.annotation</groupId>
      <artifactId>jsr250-api</artifactId>
      <version>1.0</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
<dependencies>
  <dependency>
    <groupId>org.jboss.spec.javax.faces</groupId>
    <artifactId>jboss-jsf-api_2.1_spec</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>javax.enterprise</groupId>
    <artifactId>cdi-api</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>javax.annotation</groupId>
    <artifactId>jsr250-api</artifactId>
    <scope>provided</scope>
  </dependency>
</dependencies>
</project>
```

Com nosso `pom.xml` criado, podemos importar o projeto dentro do Eclipse. Entre no seu Eclipse e selecione a opção `File > Import` na caixa de busca, digite `maven` e já deve aparecer a opção `Existing Maven Projects`. Clique em `Next` e selecione o `pom.xml`, agora basta clicar em `Finish`.

Abra a view de Servers com o atalho `Ctrl + 3` e digite `Servers`, pressione `ENTER`. Dentro da View de Servers, clique com botão direito do mouse, e vá em `New > Server`. Procure pela opção `Wildfly 10`, caso não apareça, procure por `JBoss AS, Wildfly & EAP Server Tools`. Precisamos instalar esse Tools, clicando em `Next`. Quando aparecer os termos para serem aceitos, selecione a opção `I accept...` e depois em `Finish`. Quando o processo estiver finalizado, o Eclipse irá pedir para reiniciar. Reinicie seu Eclipse.

Com o Plugin instalado, podemos colocar nosso servidor. Mais uma vez na aba `Servers`, clique com botão direito do mouse, e vá em `New > Server`. Procure pela opção `Wildfly 10` e clique em `Next` depois em `Next` novamente. Na caixa de `Home Directory`, precisaremos informar onde está a instalação do Wildfly. Clique em `Browser` e coloque a pasta descompactada do `Wildfly`. Na caixa de `Configuration file`, cliquem em `Browse...` e selecione o arquivo `standalone-full.xml`. Pressione em `Next`, então clique duas vezes no projeto `casadocodigo` que deve estar aparecendo para você, e ele estará dentro do servidor. Podemos finalizar a configuração do servidor, clicando em `Finish`.

Para que nosso projeto reconheça as bibliotecas que já estão dentro do *JBoss*, vamos precisar configurar. Clique com botão direto em cima do projeto, vá em `Build Path > Configure Build Path` selecione a aba `Libraries` e clique em `Add Library`. Procure pela opção `Server Runtime` e `Next`, depois selecione o `Wildfly 10...` que já configuramos e depois `Finish`.

Para verificarmos se realmente tudo está funcionando, precisamos criar um arquivo `index.html` dentro da pasta `webapp`. Clique na pasta `webapp` e pressione `Ctrl + N` e na caixa de busca, digite `HTML`. Vá em `Next` e dê o nome `index`. Clique em `Finish`. Abra o arquivo e vamos deixá-lo apenas com uma mensagem simples.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Casa do Código</title>
</head>
<body>

<h1>Instalação do Wildfly foi realizada com sucesso!</h1>

</body>
</html>
```

Uma vez que tudo está certinho, podemos inicializar nosso servidor. Clique na aba `Servers`, botão direito em cima do Servidor `Wildfly` e selecione `Start`.

Quando o servidor terminar de imprimir no `Console`, vá para seu navegador preferido e digite:

`http://localhost:8080/casadocodigo`. Se estiver vendo a mensagem `Instalação do Wildfly foi realizada com sucesso!`, seu projeto está configurado corretamente e já poderemos avançar.