



Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

08 - ADIÇÃO DE DEPENDÊNCIA DO JSF

O POM, cuja sigla significa **Project Object Model**, é o principal ponto de partida de um projeto que será gerenciado pelo Maven, pois é o arquivo (pom.xml) que descreve e disponibiliza as informações e configurações do projeto.

1. Configuração do JSF

1.1. Retorne ao *Eclipse* e abra o arquivo pom.xml para configurarmos a dependência do JSF.

1.2. Dentro da tag `<project>`, defina as tags abaixo (utilize `Ctrl` + `Espaço`):

```
<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>br.com.etechoracio</groupId>
  <artifactId>etector</artifactId>
  <version>0.0.1</version>
  <packaging>war</packaging>

  <dependencies>

  </dependencies>
```

1.3. Acesse o [Google](#) e procure sobre *jsf-api javax.faces maven*.

1.4. Entre na página do **Maven Repository** e escolha a opção **2.2.2**.

1.5. Copie o texto da dependência:

Maven Gradle SBT Ivy Grape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/com.sun.faces/jsf-api -->
<dependency>
  <groupId>com.sun.faces</groupId>
  <artifactId>jsf-api</artifactId>
  <version>2.2.2</version>
</dependency>
```

1.6. Cole-o dentro da tag `<dependencies>` do pom.xml.

1.7. Repita a operação para a versão **2.2.2** do *jsf-impl*.

1.8. Pressione `Ctrl` + `Shift` + `F` para indentar o arquivo e salve-o.



Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

- 1.9. Atualize o projeto, clique com o botão direito no projeto e acesse **Maven** → **Update Project**.
- 1.10. Clique com botão direito no projeto e acesse a opção **Properties**.
- 1.11. Selecione o menu **Project Facets**, desmarque a opção **JavaServer Faces** e clique em **Apply**.
- 1.12. Torne a marcar a opção **JavaServer Faces** e observe que apareceu a opção **Further configuration available**:



- 1.13. Ao clicar em **Further configuration available**, em **Type**, selecione a opção **Disable Library Configuration**.
- 1.14. Ainda nessa tela, em **URL Mapping**, remova o mapeamento **/faces/*** e adicione um novo mapeamento como ***.xhtml**.
- 1.15. Clique em **Ok** e depois em **Apply and Close**.
- 1.16. O erro que havia no `pom.xml` foi corrigido.
- 1.17. Acesse a pasta **C:/Java-Projeto/etector** (via **Prompt de Comando**).
- 1.18. Verifique a situação dos arquivos no repositório Git.

```
git status
```

- 1.19. Faça com que os arquivos sejam rastreados pelo Git.

```
git add .
```

- 1.20. Verifique a situação dos arquivos no repositório Git novamente.
- 1.21. Execute o **commit** para gravar as mudanças no repositório com a mensagem **“Configuração do JSF”**.
- 1.22. Efetue um **push** para enviar os arquivos ao repositório remoto.
- 1.23. Abra a página web do **Github** e verifique se o arquivo `pom.xml` foi atualizado (seção **commits**).