

FIAP GRADUAÇÃO

# DIGITAL BUSINESS ENABLEMENT

Prof. MSc. Rafael Matsuyama

#05 – WEB SERVICES SOAP - IMPLEMENTAÇÃO



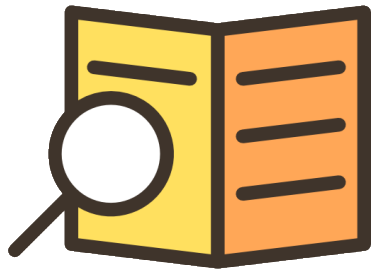
# PERCURSO

---



## #05 - AGENDA

---



- Web Service Provider
  - Configuração do eclipse para Axis 2
  - Criação do projeto
  - Gerar um serviço web
- Web Service Requester
  - Interface Texto
    - Criação do projeto
    - Geração das classes de acesso ao web service
    - Exemplo de acesso ao web service
  - Interface Desktop
    - Instalação do plugin no eclipse
    - Criação do projeto
    - Componentes de tela

- **JAX WS** – JAVA API FOR XML WEB SERVICES
- API do **JAVA EE** para criação de web services;
- **Implementações:**
  - Apache Axis (JAX-RPC)
  - **Apache Axis 2 (JAX-WS, JAX-RS)**
  - Apache CXF (JAX-WS, JAX-RS)
  - Java SE Runtime (a partir do Java 6) (JAX-WS)



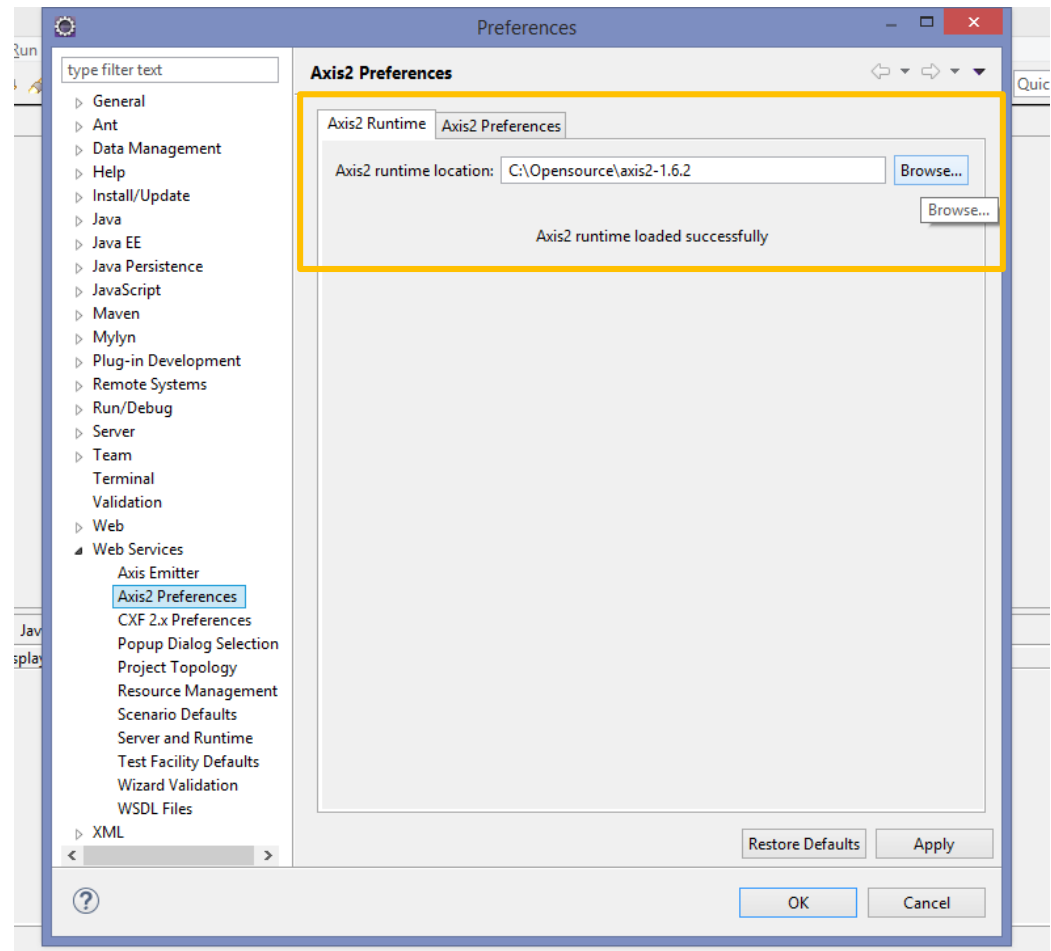
# WEB SERVICE PROVIDER

# CONFIGURAÇÃO DO ECLIPSE

- Faça o download do Apache Axis 2 (utilize as versões 1.6.x) através do link:

<http://axis.apache.org/axis2/java/core/download.cgi>

Em **Windows > Preferences**, procure por **Axis2 Preferences** e configure a localização do apache axis 2.



# CRIANDO O PROJETO

- 1 - Crie um **Dynamic Web Project**;
- 2 - Configure o Target Runtime: **Apache Tomcat 9**;
- 3 - Mude o **Dynamic web module version: 2.5**;
- 4 e 5 - Em configuration clique em : **Modify** > Marque: **Axis 2 Web Services**;

The left screenshot shows the 'New Dynamic Web Project' dialog. It has a title bar with a gear icon and window controls. The main title is 'Dynamic Web Project'. Below it is a description: 'Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.' There are four numbered callouts: 1 points to the 'Project name' field (containing 'ExemploWs'), 2 points to the 'Target runtime' dropdown (showing 'Apache Tomcat v8.0'), 3 points to the 'Dynamic web module version' dropdown (showing '2.5'), and 4 points to the 'Modify' button at the bottom right.

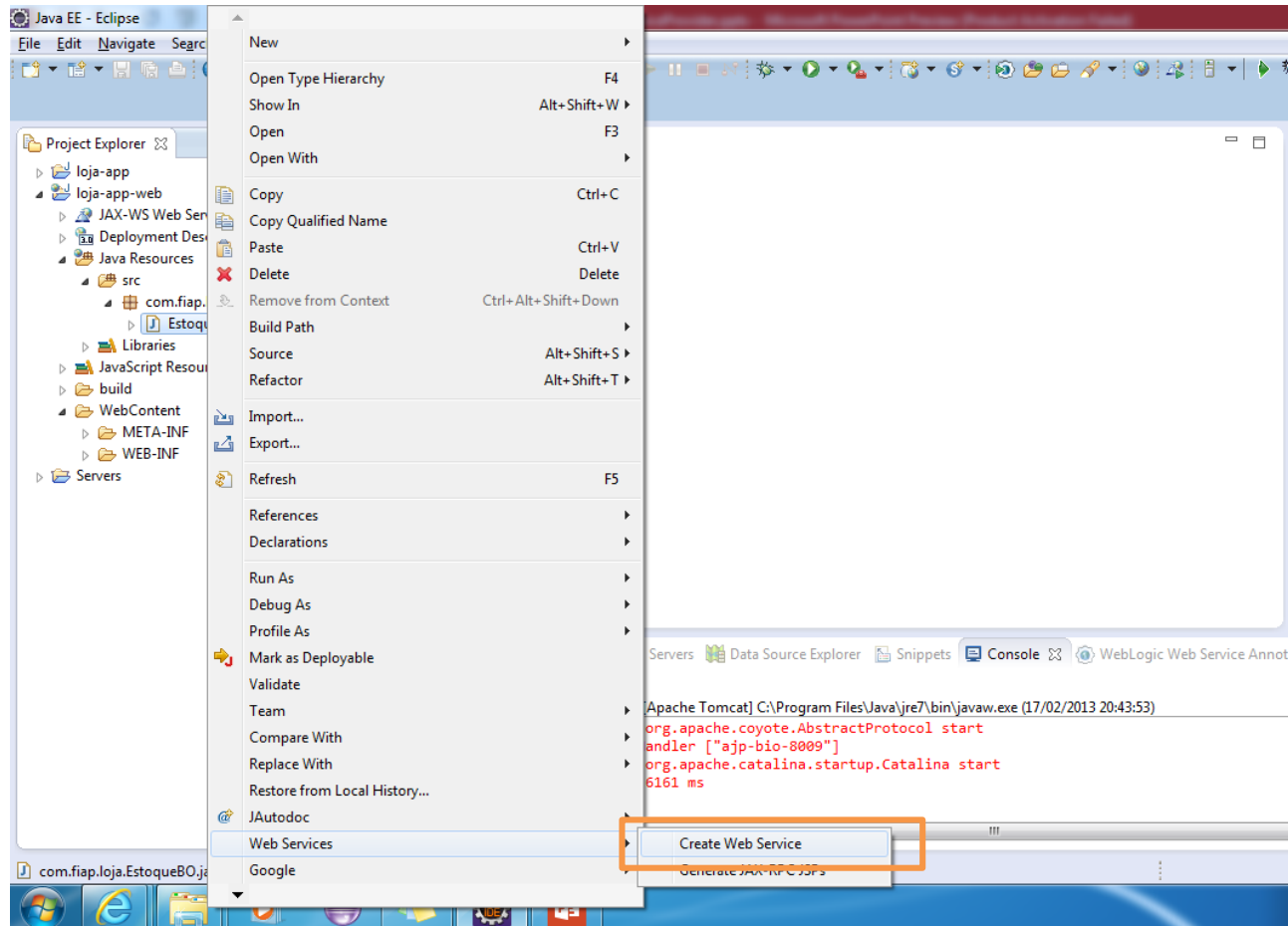
The right screenshot shows the 'Project Facets' dialog. It has a title bar with a gear icon and a 'Project' label. The main title is 'Project Facets'. Below it is a description: 'Select the facets that should be enabled for this project.' There is a 'Configuration:' dropdown showing '<custom>'. Below that is a table with 'Project Facet' and 'Version' columns. The table has five numbered callouts: 5 points to the 'Axis2 Web Services' row, which is checked. The other rows are unchecked.

Project Facet	Version
<input checked="" type="checkbox"/> Axis2 Web Services	
<input type="checkbox"/> CXF 2.x Web Services	1.0
<input checked="" type="checkbox"/> Dynamic Web Module	2.5
<input checked="" type="checkbox"/> Java	1.8
<input type="checkbox"/> JavaScript	1.0
<input type="checkbox"/> JavaServer Faces	2.2
<input type="checkbox"/> JAX-RS (REST Web Services)	1.1
<input type="checkbox"/> JAXB	2.2
<input type="checkbox"/> JPA	2.1
<input type="checkbox"/> WebDoclet (XDoclet)	1.2.3

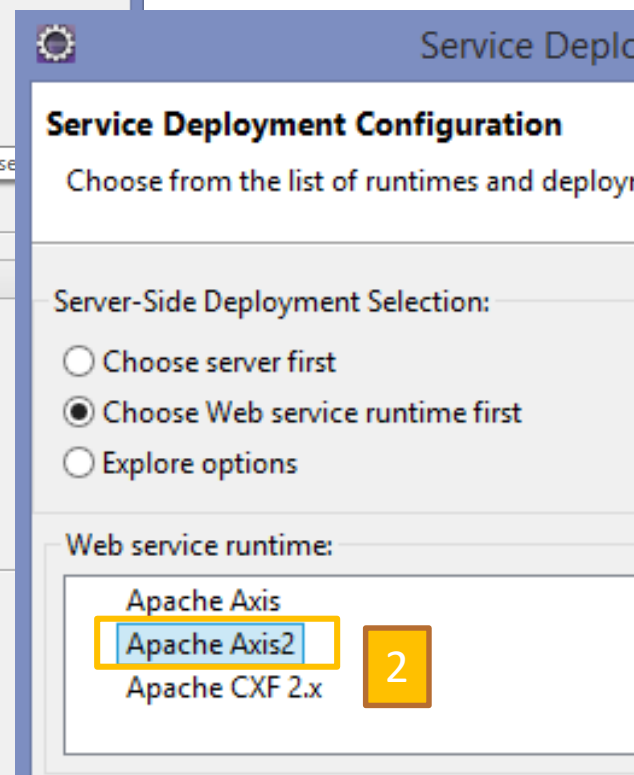
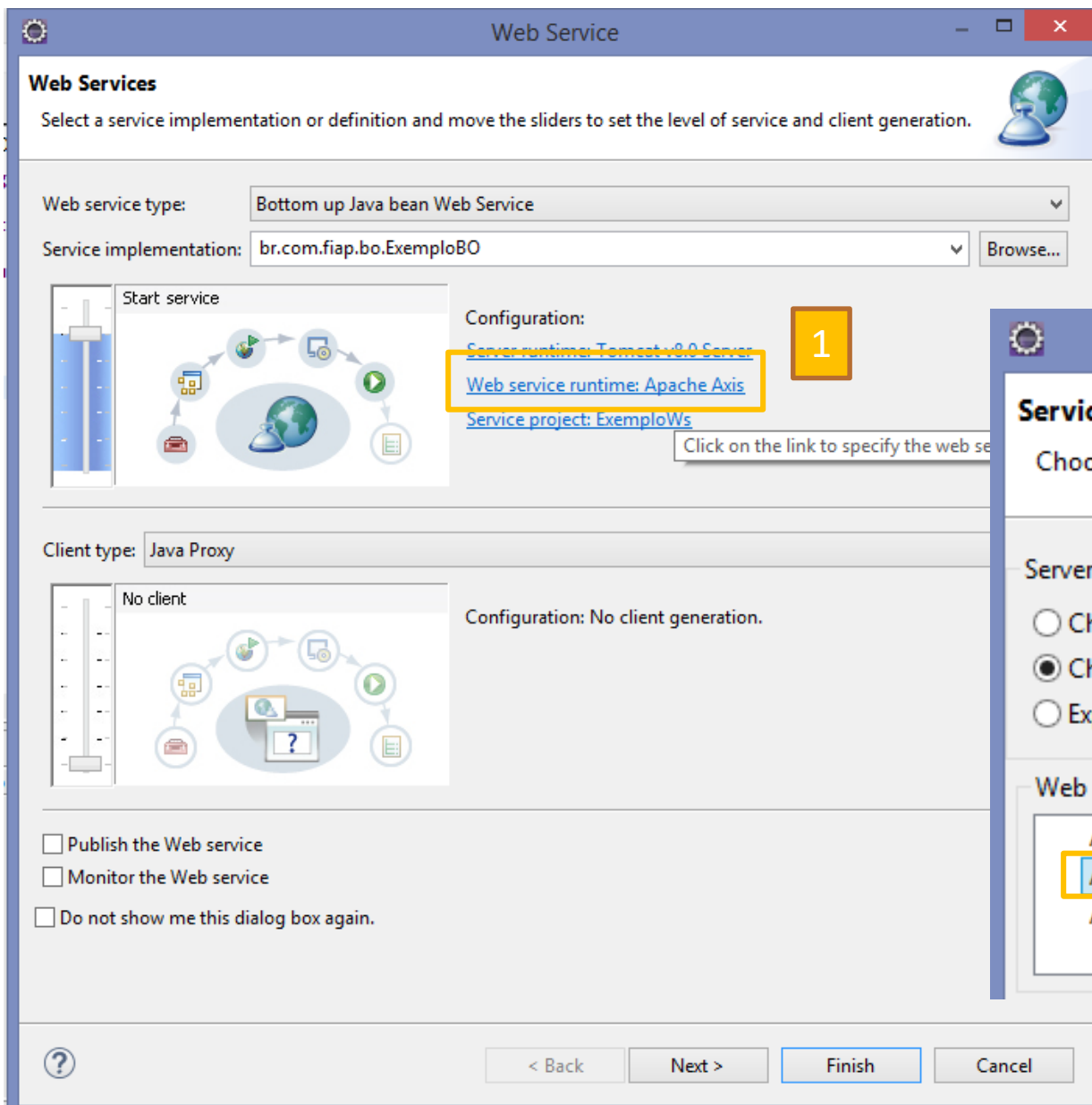


# GERANDO UM SERVIÇO WS

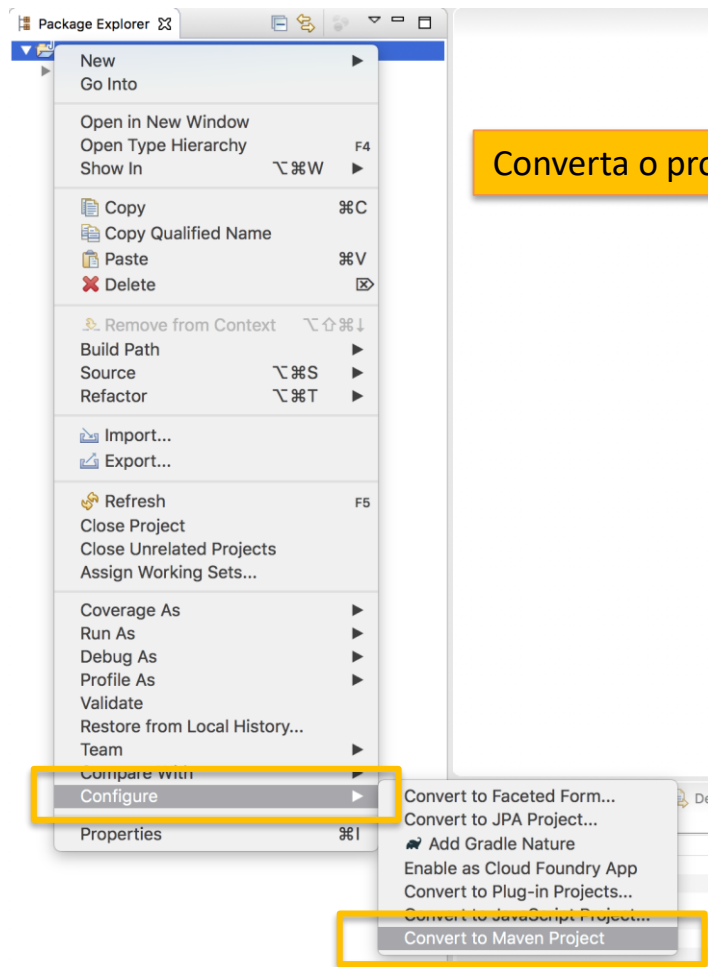
- Clique com o botão direito na classe e escolha **Web Services> Create Web Service;**



# GERANDO UM SERVIÇO WS



- Em alguns casos, é necessário colocar a dependência do Axis2 diretamente no pom.xml, neste caso:



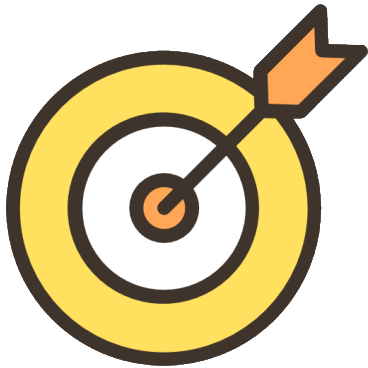
Converte o projeto para **Maven**

```
<dependencies>
  <dependency>
    <groupId>org.apache.axis2</groupId>
    <artifactId>axis2-adb</artifactId>
    <version>1.6.4</version>
  </dependency>
</dependencies>
```

Adicione a dependência no pom.xml

# PRÁTICA!

---

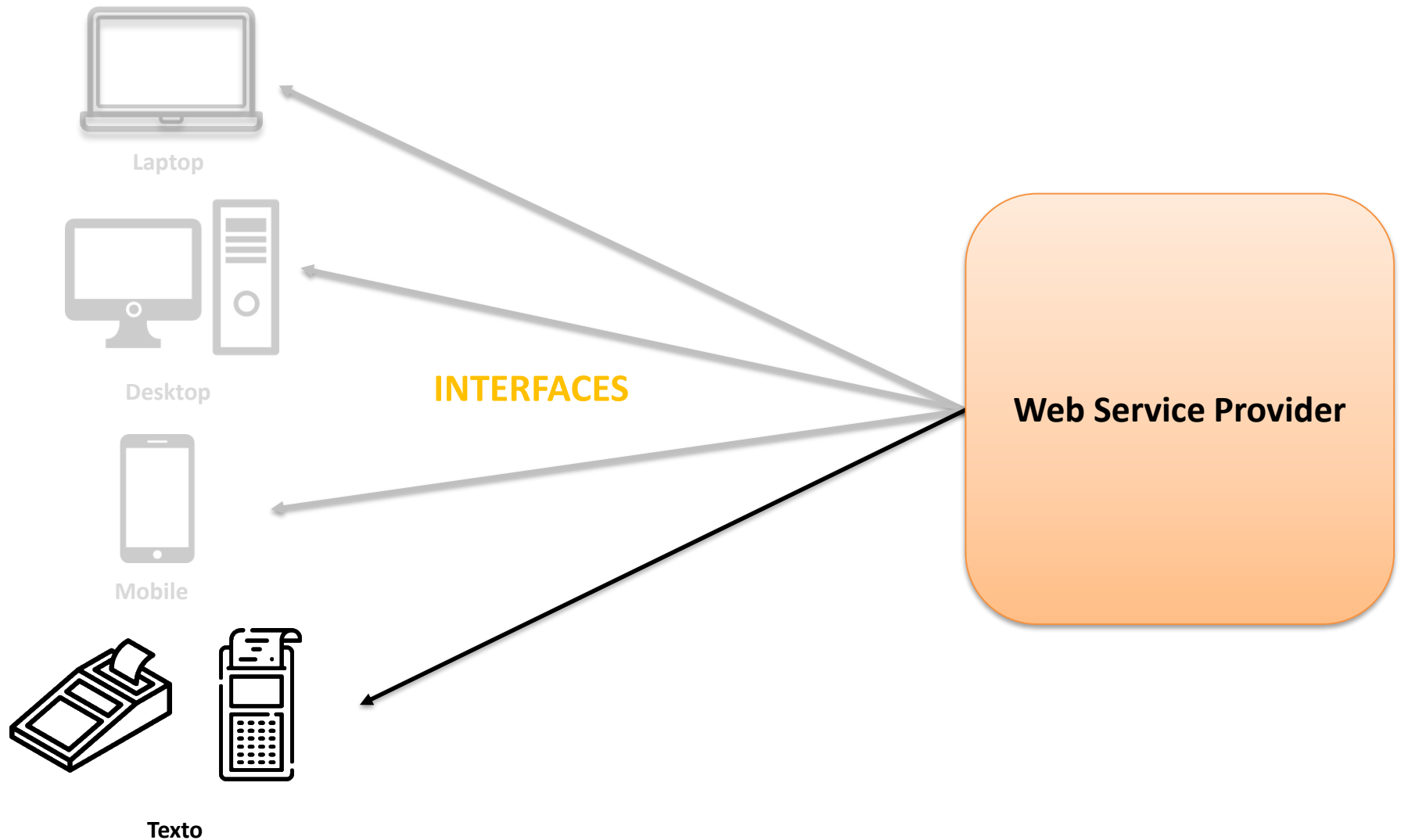


1. Criar um projeto **Java Web Application** com o **Axis 2**;
2. Desenvolver um **Web Services Provider** para obter informações a respeito de **produtos** de uma loja. **(Utilize o projeto do exercício anterior (Apostila 03))**;
3. Gerar um **AxisFault** caso o produto não esteja cadastrado (**throw new AxisFault("Produto Não Cadastrado")**);
4. Teste o serviço com o **SOAP UI**;



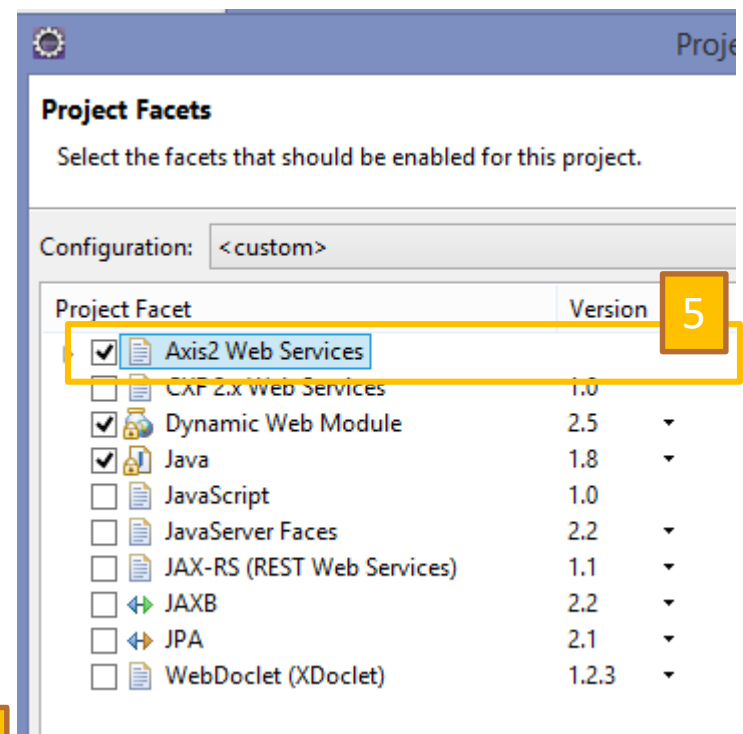
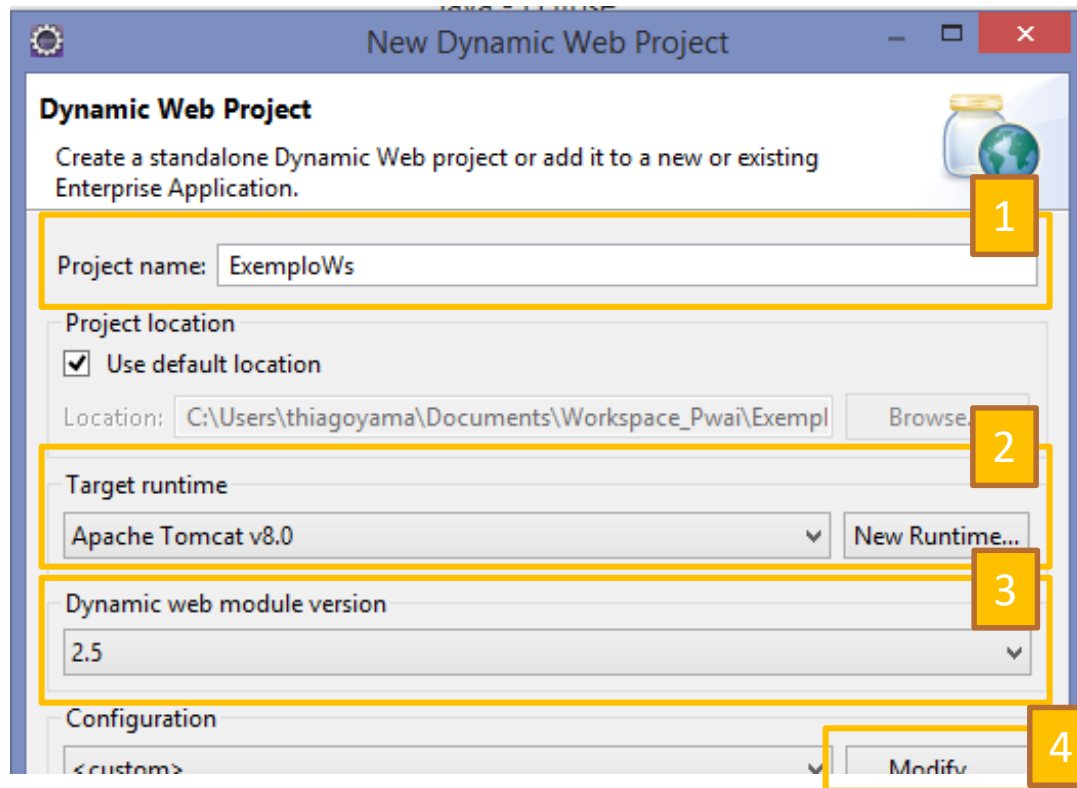


# WEB SERVICE REQUESTER



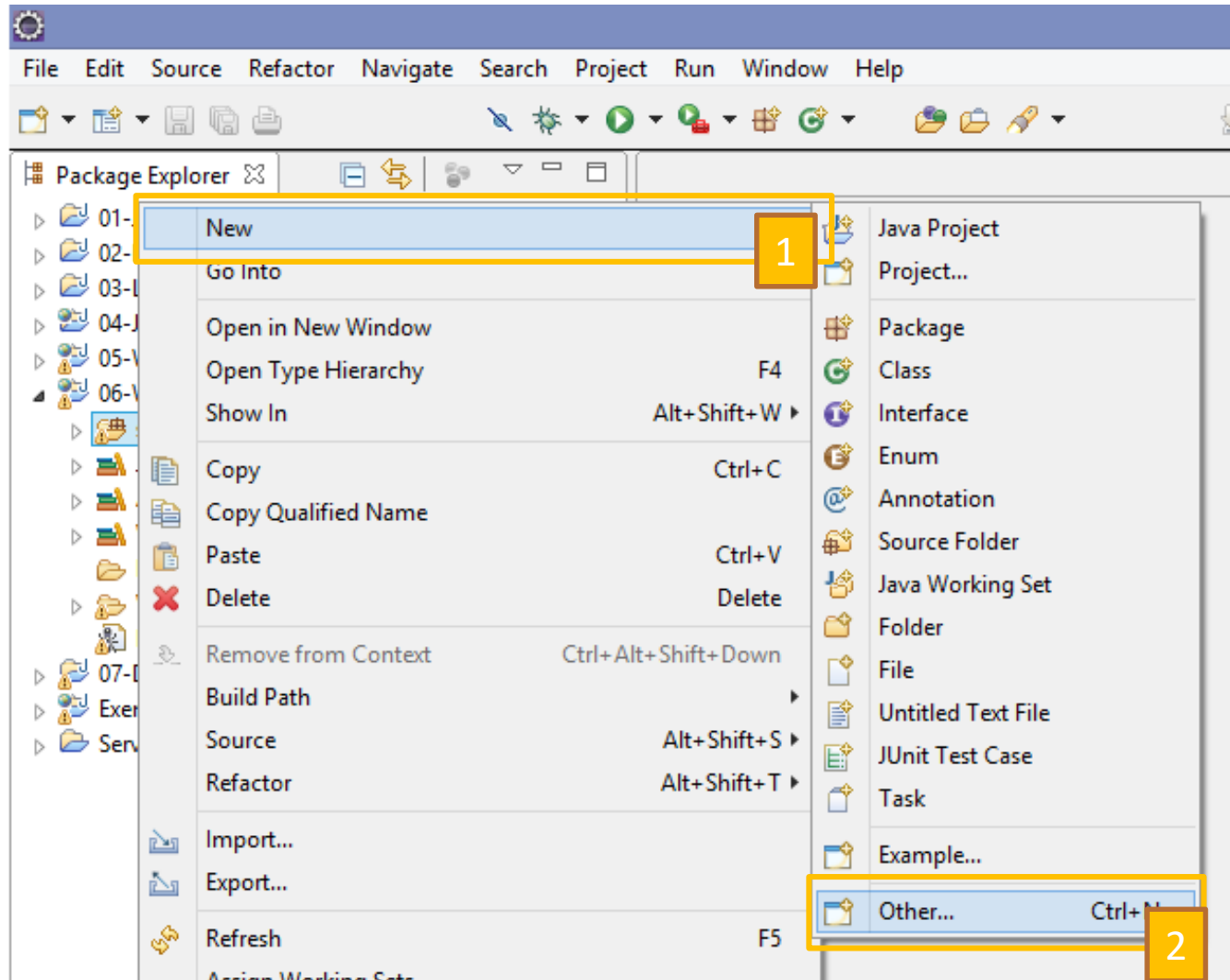
# PROJETO – WS PROVIDER

- Crie um **Dynamic Web Project**;
- Configure o Target Runtime: **Apache Tomcat 9**;
- Mude o **Dynamic web module version: 2.5**;
- Em configuration clique em : **Modify** > Marque: **Axis 2 Web Services**;



# CRIANDO O WEB SERVICE CLIENT

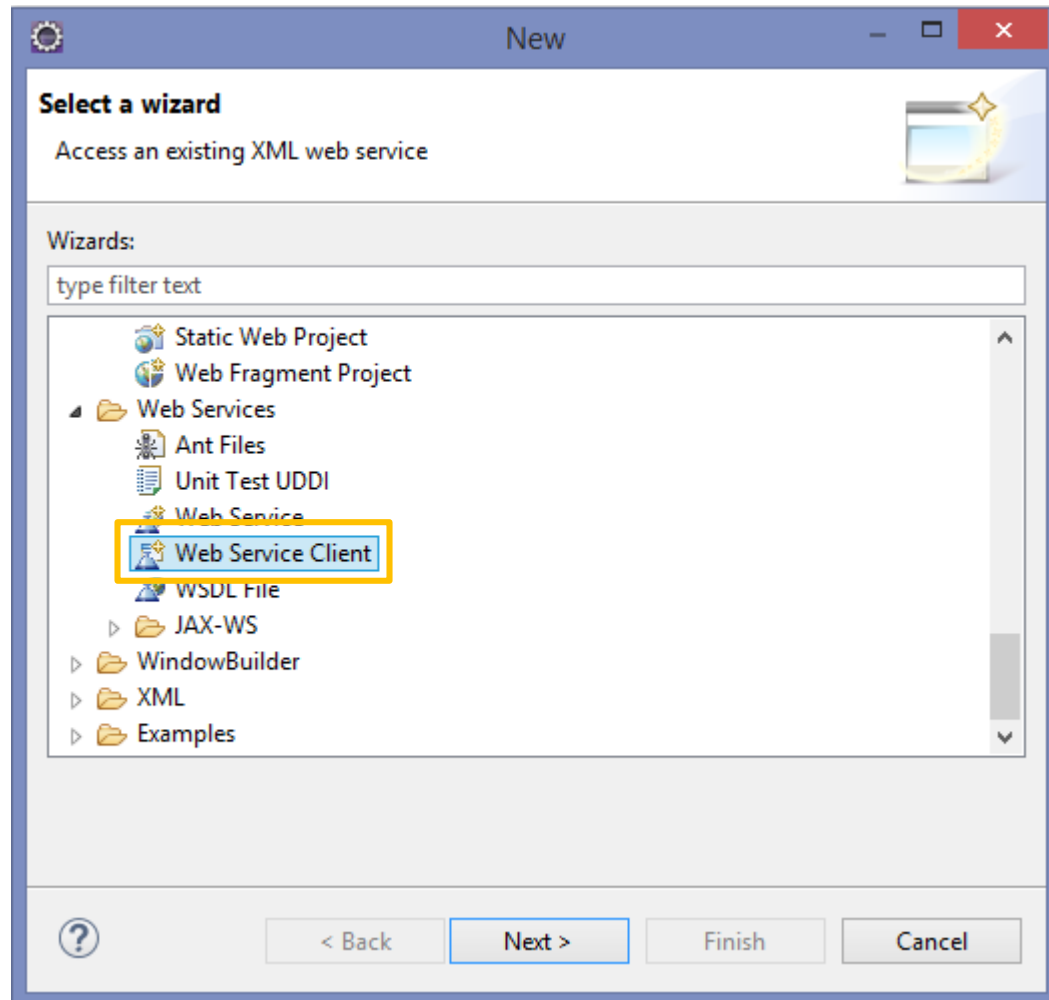
- Clique com o botão direito do mouse no projeto e escolha **New > Other...**





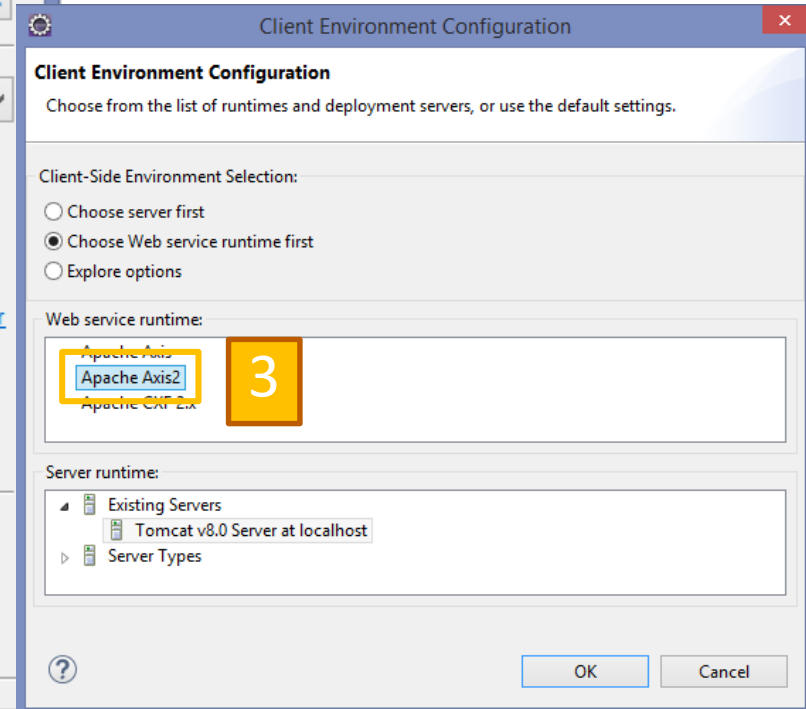
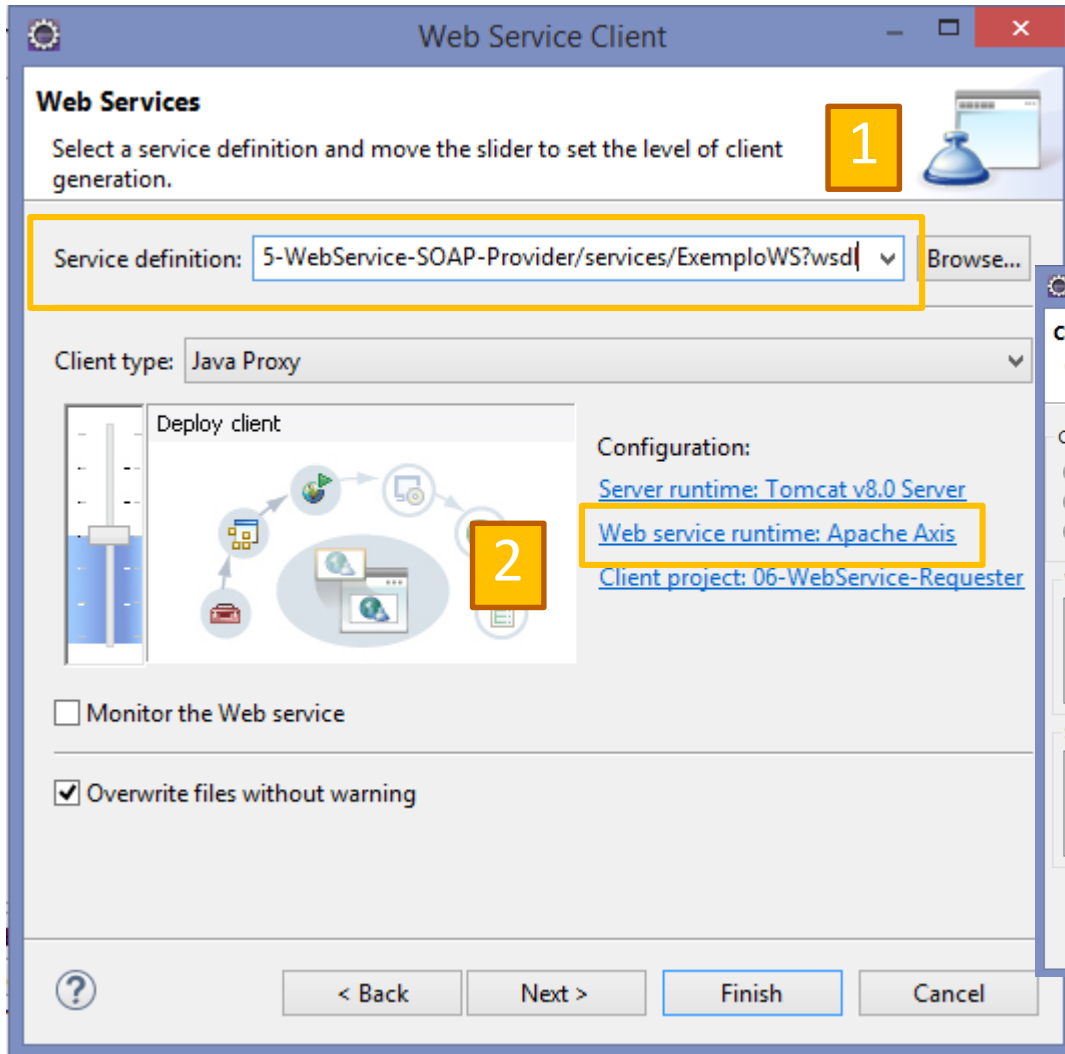
# CRIANDO O WEB SERVICE CLIENT

- Escolha a opção **Web Service Client**



# CRIANDO O WEB SERVICE CLIENT

- Coloque o endereço do WSDL e ajuste o Apache Axis 2;



- Instancie o **Stub** e para cada método do web service, instancie uma classe com o **mesmo nome do método** para enviar valores ao ws.

```
ExemploWSStub stub = new ExemploWSStub();
```

```
//Cadastrar um carro
```

```
CarroTO param = new CarroTO();
```

```
param.setModelo("GOL");
```

```
param.setPreco(10000);
```

```
Cadastrar cadastrar = new Cadastrar();
```

```
cadastrar.setCarro(param);
```

```
stub.cadastrar(cadastrar);
```

```
//Listar carros
```

```
Listar listar = new Listar();
```

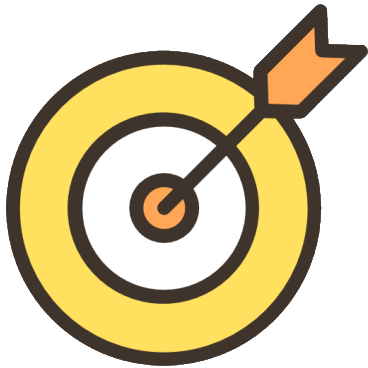
```
ListarResponse response = stub.listar();
```

```
CarroTO[] vetor = response.get_return();
```

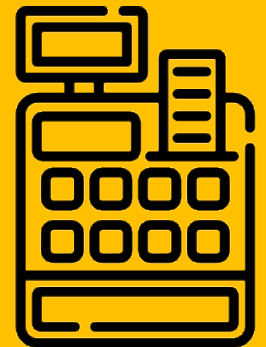
```
List<Carro> lista = Arrays.asList(vetor);
```

## PRÁTICA! INTERFACE TEXTO

---

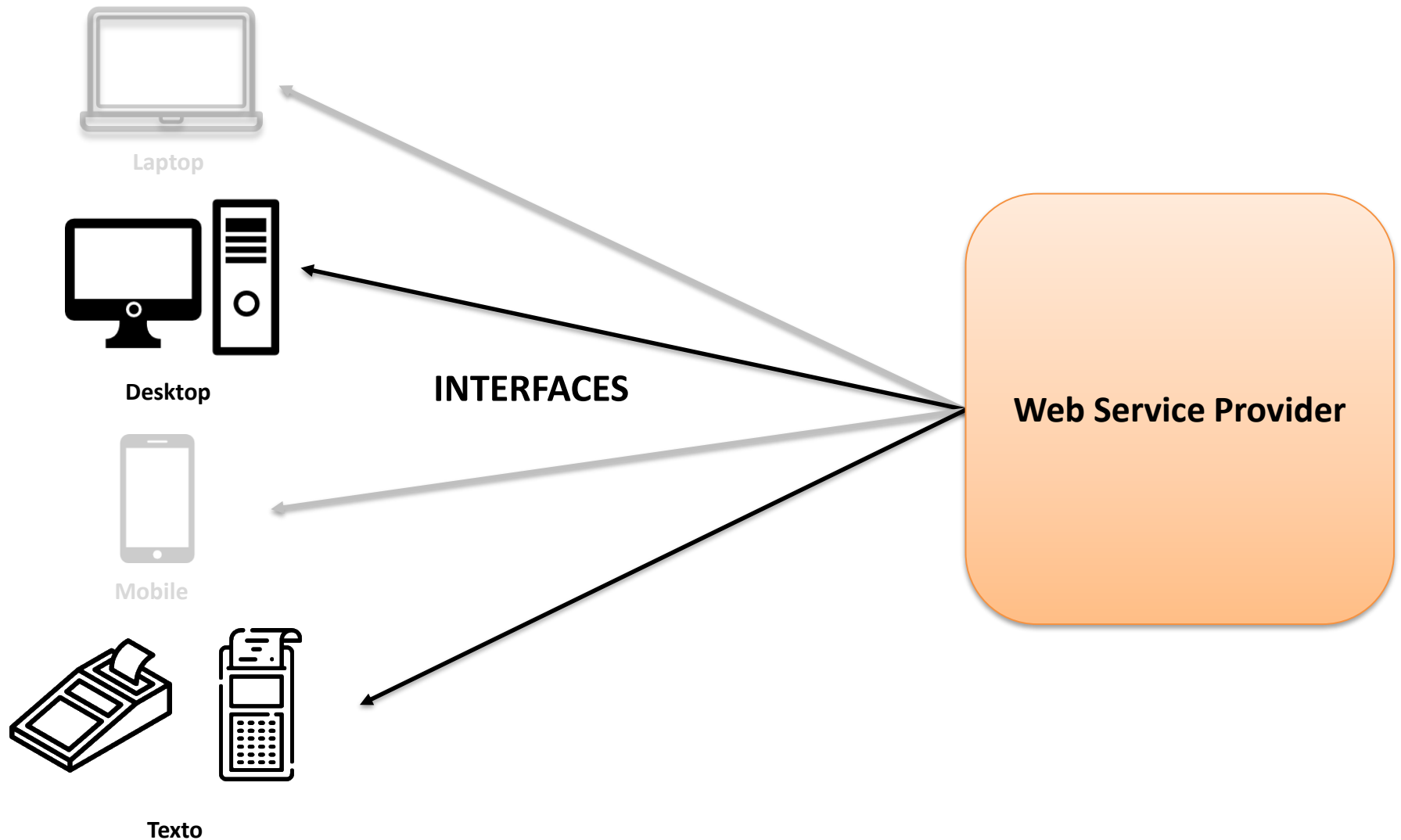


1. Criar um projeto para o **Web Services Requester**;
2. Criar as **classes de acesso** ao Web Service do exercício anterior;
3. Implemente uma classe com o método **main** para que o **usuário possa informar o código** de um produto e o programa **acesse o web service** para obter a resposta;





# DESKTOP

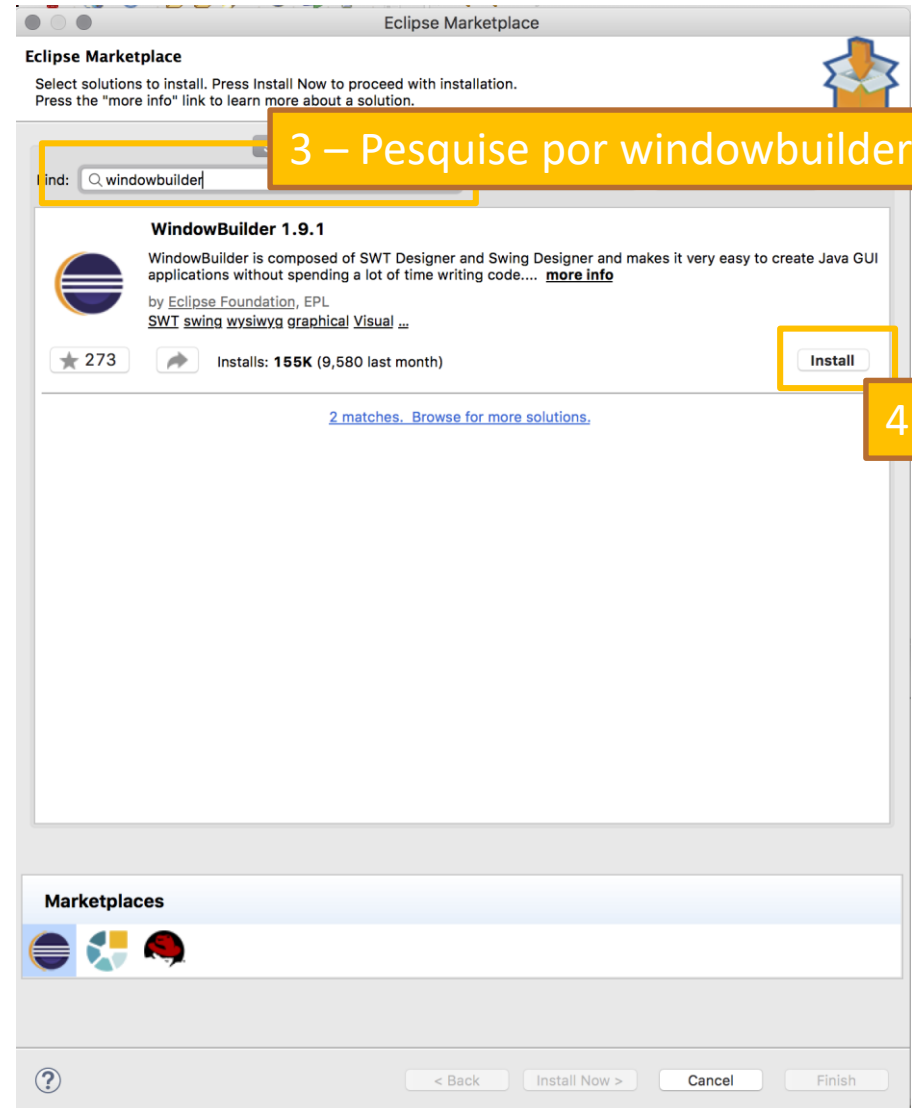
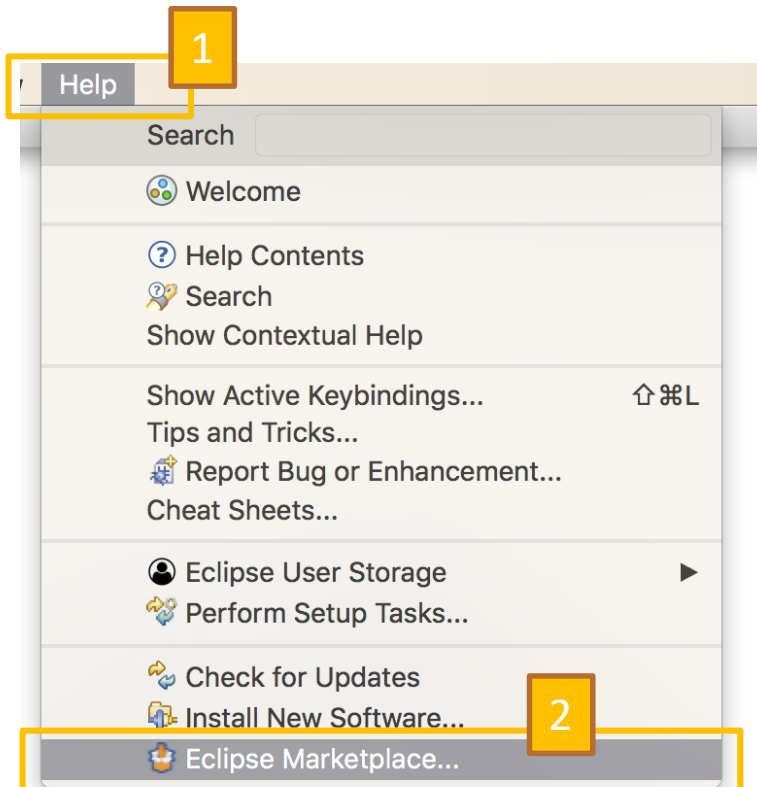


- Para desenvolver aplicações **desktop** no eclipse podemos utilizar **Swing, AWT** e etc..;
- Nós vamos utilizar o **SWT: The Standard Widget Toolkit**;
- Para isso, precisamos instalar um **plugin**;



# PLUGIN NO ECLIPSE

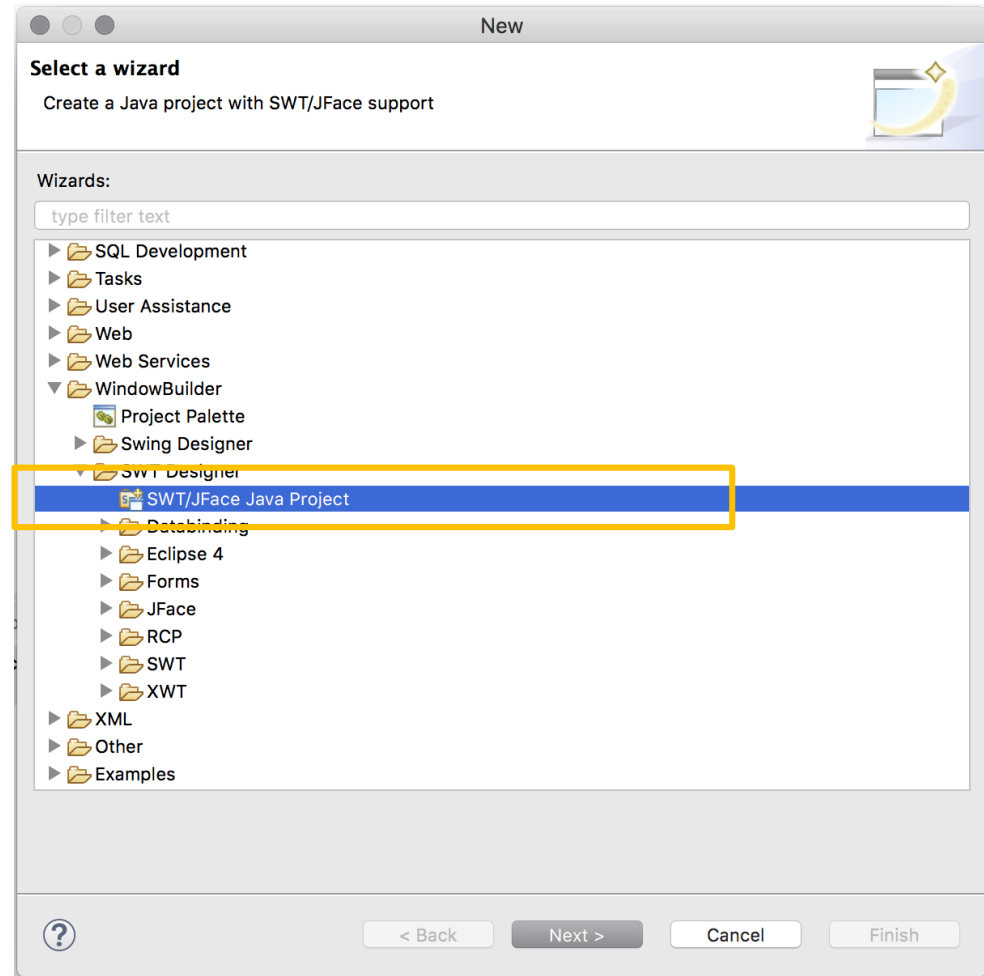
- Uma característica do eclipse é a possibilidade para **instalar plugins** para aumentar as funcionalidades da IDE:





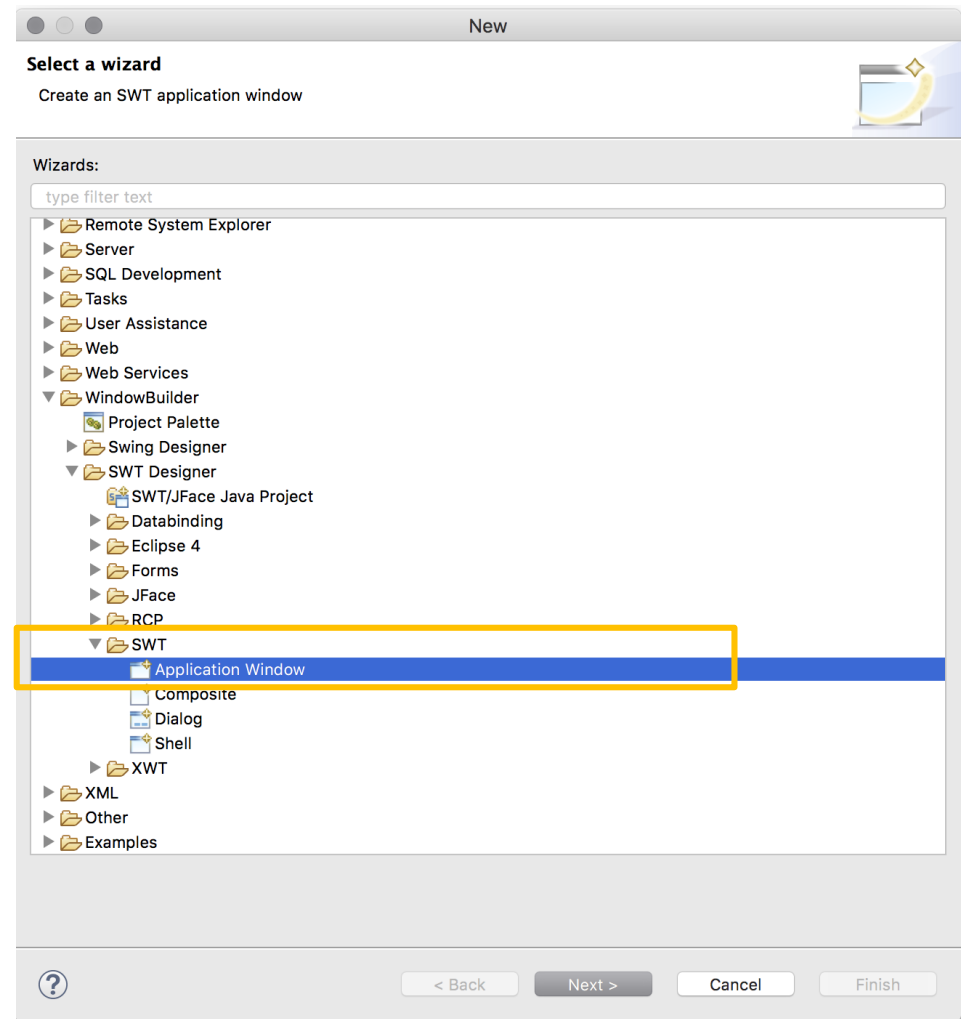
# CRIAÇÃO DO PROJETO

- Para criar um projeto Desktop, escolha a opção **File > New > Other..**
- Dentro da pasta **WindowBuilder**, escolha a opção **SWT/Jface Java Project**;

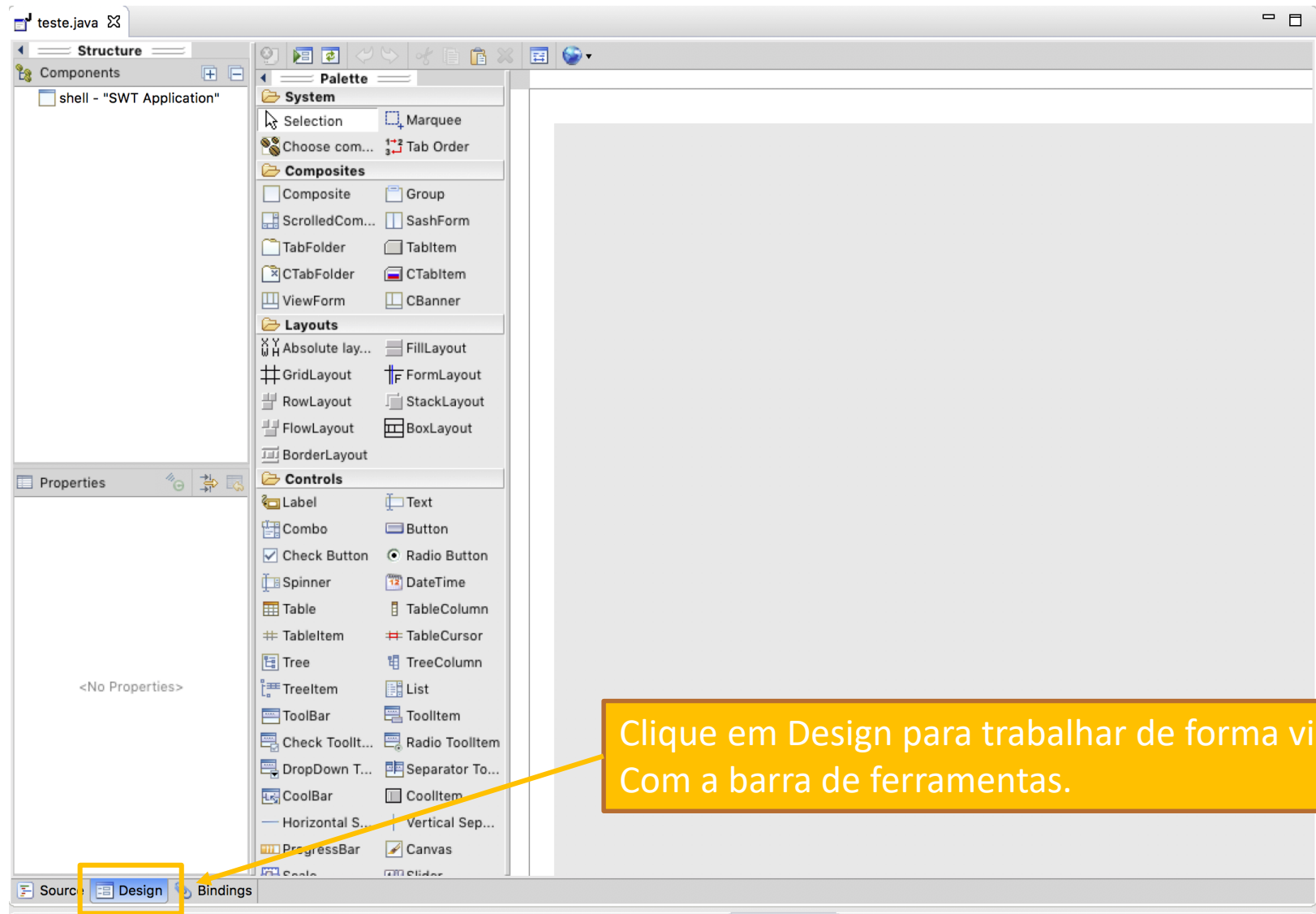


# CRIAÇÃO DE UMA TELA

- Dentro do projeto, crie um **Application Window** que está dentro da pasta **SWT**;
- Será criado uma **classe** que **renderiza** uma **tela** dentro da aplicação;



# BARRA DE FERRAMENTAS



- Controls são objetos que permitem a interação com a interface preparada para o usuário;
- Estes controls podem ser instanciados visualmente utilizando o WindowBuilder;
- **Label**: Apresenta dados na tela
  - Propriedade **Font** permite alterar aparência visual do objeto;
  - Propriedade **Foreground** permite alterar cor do texto;
  - Propriedade **Image** permite você incluir uma imagem no objeto;
  - Selecione o objeto -> botão direito -> Autosize Control para ajustar texto;
- **Text**: Permite apresentação e edição de dados
  - Propriedade **Variable** permite definir um nome para o objeto;
  - Prefixo padrão de nome **"txt"**;
  - Propriedade **Editable** permite definir se o campo é de consulta ou para preenchimento de dados;
  - Método **getText()** permite recuperar o texto digitado pelo usuário;
  - Método **setText()** permite alterar o conteúdo texto de um objeto na tela;

- **Table**: Apresenta dados em formato de tabela;
  - Propriedade **Variable** define o nome do objeto a ser instanciado;
  - Prefixo padrão de nome **"tbl"**;
- **TableItem**: Filho de Table.
  - Representa as colunas de uma tabela;
  - Propriedade **Text** define nome das colunas;
  - Para incluir um novo item de forma programática basta instanciar um objeto do tipo **TableItem**;
  - A sequência das colunas deve obedecer a sequência de inserção:

```
TableItem item = new TableItem(table, 0);  
item.setText(new String[] {"1", "Calça"});
```

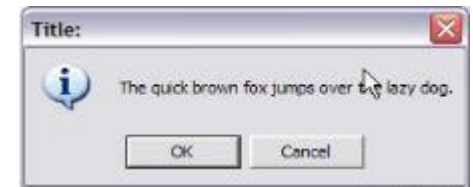
- **Button**: Ação de usuário por meio de botão;
- **Prefixo** padrão de nome **“btn”**;
- Para incluir uma **ação no botão** dê dois cliques no botão ou selecione o objeto > botão direito > Add Event Handler > Selection > widgetSelected;

```
Button b = new Button(shell,SWT.PUSH);
b.setText("OK");
b.addSelectionListener(new SelectionAdapter() {
    public void widgetSelected(SelectionEvent e) {
        MessageDialog.openInformation(shell, "SWT", "Ola Mundo!");
    }
});
```

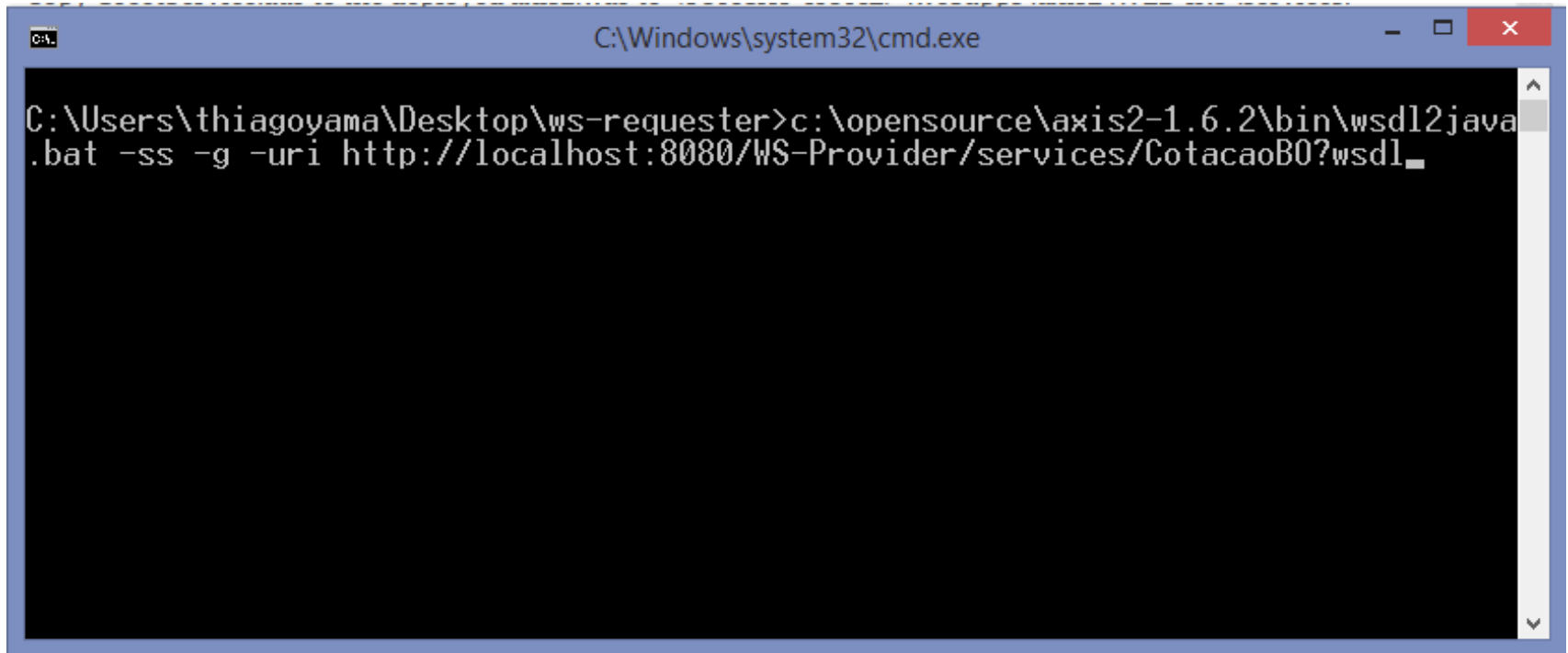
# | MENSAGEM DE CONFIRMAÇÃO

- O componente abre uma **caixa de confirmação**, o retorno será a constante que representa o botão que o usuário clicou;

```
MessageBox mb = new MessageBox(shell, SWT.OK|SWT.CANCEL);  
mb.setMessage("Clique OK caso queira encerrar a aplicação");  
int result = mb.open();  
if ( result == SWT.OK) {  
    System.out.println("OK foi pressionado");  
    System.exit(0); /* encerra programa  
}  
if (result == SWT.CANCEL)  
    System.out.println("cancela foi pressionado");
```



- O projeto **Desktop** não consegue gerar os códigos para acessar o ws, dessa forma, utilize uma ferramenta do axis2 para gerar as classes:



```
C:\Windows\system32\cmd.exe

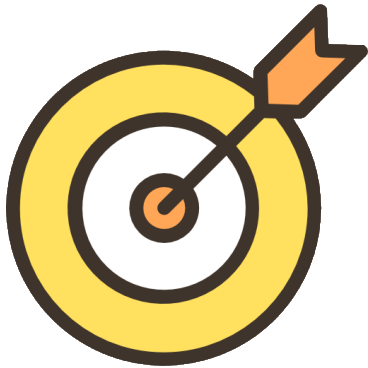
C:\Users\thiagoyama\Desktop\ws-requester>c:\opensource\axis2-1.6.2\bin\wsdl2java
.bat -ss -g -uri http://localhost:8080/WS-Provider/services/CotacaoB0?wsdl_
```

- Também é possível utilizar as classes gerada em projetos anteriores;



## PRÁTICA! INTERFACE DESKTOP

---



1. Criar um projeto **Desktop**;
2. Utilize as **classes de acesso** ao Web Service do exercício anterior;
3. Implemente uma **tela** para realizar a **pesquisa de produtos** e a **listagem de produtos**;
4. Faça o **deployment** da aplicação (exe).



## VOCÊ APRENDEU...

---



- Implementar um **web service provider** com Axis 2;
- Criar um **web service requester** com **interface texto**;
- Desenvolver uma aplicação **desktop** para consumir um **web service SOAP**;

**Copyright © 2018 – 2019**

**Prof. MSc. Rafael Matsuyama / Prof. Me. Thiago T. I. Yamamoto**

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

*“Nosso maior medo não deve ser o fracasso, mas  
ser bem-sucedidos em algo que não importa”*