

PROYECTO DE IMPLEMENTACIÓN CENTRADA EN ARQUITECTURA

Arquitectura de Servicios

Ing. Leonardo Cruz Mesa

Arquitecto SOA

Especialista en Arquitectura Empresarial de
Software

Agenda

1. Preguntas del Proyecto.
2. Tareas PICA
3. Generalidades SOA.
4. Generalidades de los Web Services.
5. Taller WebServices JAX-WS.
 - a) Top-Down Proyecto Sony.
 - b) Bottom-up Proyecto Sony.
6. Arquitecturas Web.
7. Ejercicios Adicionales
 1. JSP proyecto Date.
 2. JSF proyecto GuessNumber.

Preguntas del Proyecto

Tareas PICA

1. Quiz Miércoles 9 de Agosto- Temas : SOA, XML, Anatomía Servicio.
2. Trabajo de ajustar los contratos y validarlos para:
 - Pasarela de Pagos (MasterCard y Visa)
 - Rapid Service
 - DHL

Entrega Miércoles 9 de Agosto a media noche por empresa PICA en la carpeta (partners WSDL):

La entrega debe ser WSDL por cada empresa proveedora de KallSonys.
La validación del contrato se realizara con SOAPUI.

Generalidades SOA

SOA

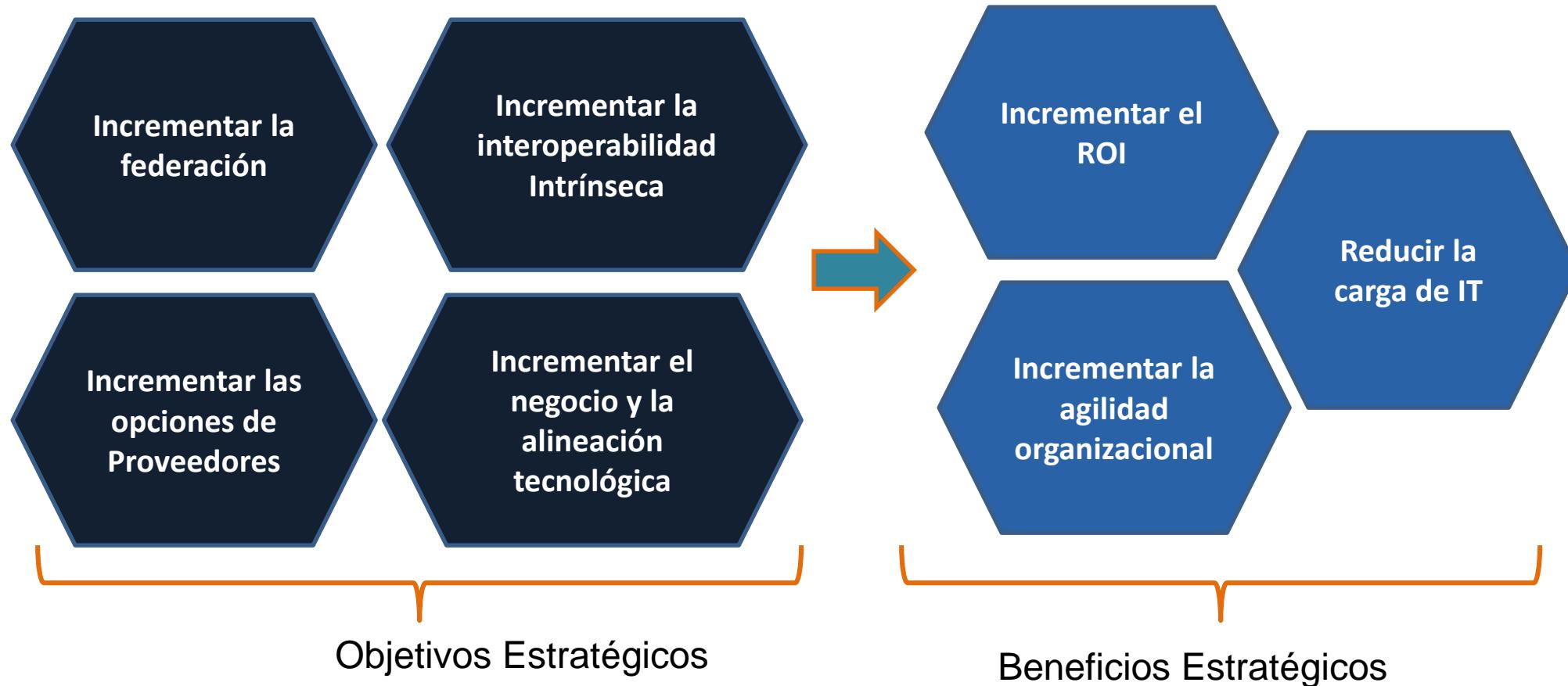
Representa un modelo de arquitectura que apunta a obtener agilidad, rentabilidad en la empresa mientras se reduce la sobrecarga en la organización

Se logra mediante implementación de servicios como medio principal para representar la lógica de una solución

SOA soporta la orientación a servicios en la realización de los objetivos estratégicos asociados a la computación orientada a servicios



SOA – Siete Objetivos Estratégicos



SOA – Principios SOA

Contratos
estandarizados

Bajo
acoplamiento

Abstracción

Reutilización

Autonomía

Ausencia de
estado

Capacidad de
ser
descubiertos

Capacidad de
ser
combinados



SOA – Patrones Diseño SOA

| | | | |
|----------------------------------|----------------------------|--------------------------|-------------------------------|
| Agnostic Capability | Contract Centralization | Functional Decomposition | Proxy Capability |
| Agnostic Context | Contract Denormalization | Idempotent Capability | Redundant Implementation |
| Agnostic Sub-Controller | Cross-Domain Utility Layer | Intermediate Routing | Reference Data Centralization |
| Asynchronous Queuing | Data Confidentiality | Inventory Endpoint | Reliable Messaging |
| Atomic Service Transaction | Data Format Transformation | Legacy Wrapper | Reusable Contract |
| Augmented Protocols | Data Model Transformation | Lightweight Endpoint | Rules Centralization |
| Brokered Authentication | Data Origin Authentication | Logic Centralization | Schema Centralization |
| Canonical Expression | Decomposed Capability | Message Screening | Service Agent |
| Canonical Protocol | Decoupled Contract | Messaging Metadata | Service Callback |
| Canonical Resources | Direct Authentication | Metadata Centralization | Service Data Replication |
| Canonical Schema | Distributed Capability | Micro Task Abstraction | Service Decomposition |
| Canonical Versioning | Domain Inventory | Microservice Deployment | Service Encapsulation |
| Capability Composition | Dual Protocols | Multi-Channel Endpoint | Service Façade |
| Capability Recomposition | Endpoint Redirection | Non-Agnostic Context | Service Grid |
| Compatible Change | Enterprise Inventory | Partial State Deferral | Service Instance Routing |
| Compensating Service Transaction | Entity Abstraction | Partial Validation | Service Layers |
| Composition Autonomy | Entity Linking | Policy Centralization | Service Messaging |
| Concurrent Contracts | Event-Driven Messaging | Process Abstraction | Service Normalization |
| Containerization | Exception Shielding | Process Centralization | Service Perimeter Guard |
| Content Negotiation | File Gateway | Protocol Bridging | Service Refactoring |
| State Messaging | Stateful Services | Trusted Subsystem | Utility Abstraction |
| State Repository | Termination Notification | UI Mediator | Validation Abstraction |
| | | | Version Identification |

SOA – Patrones Diseño SOA

Canonical
Schema Bus

Enterprise
Service Bus

Federated
Endpoint Layer

Official Endpoint

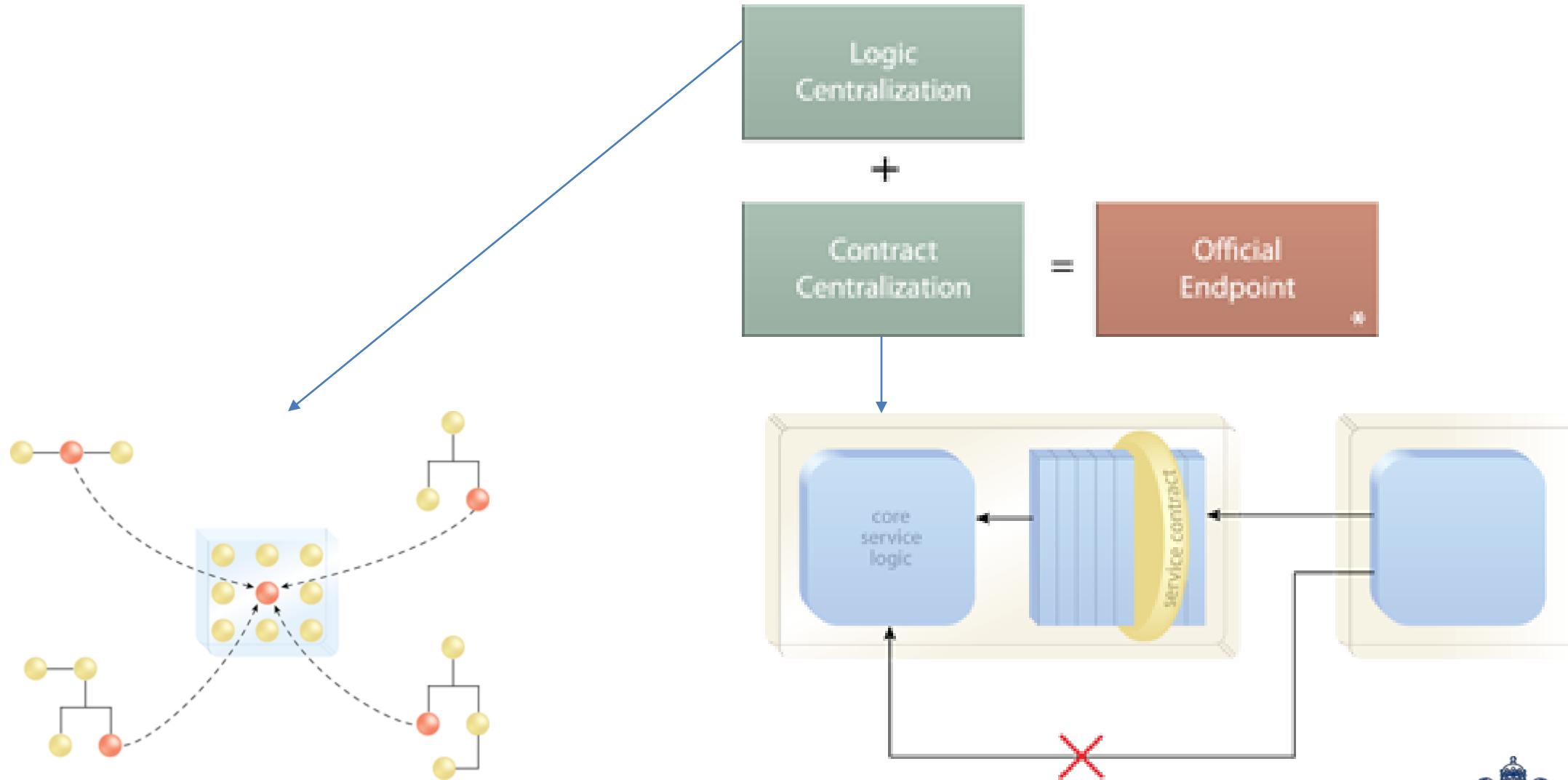
Orchestration

Service Broker

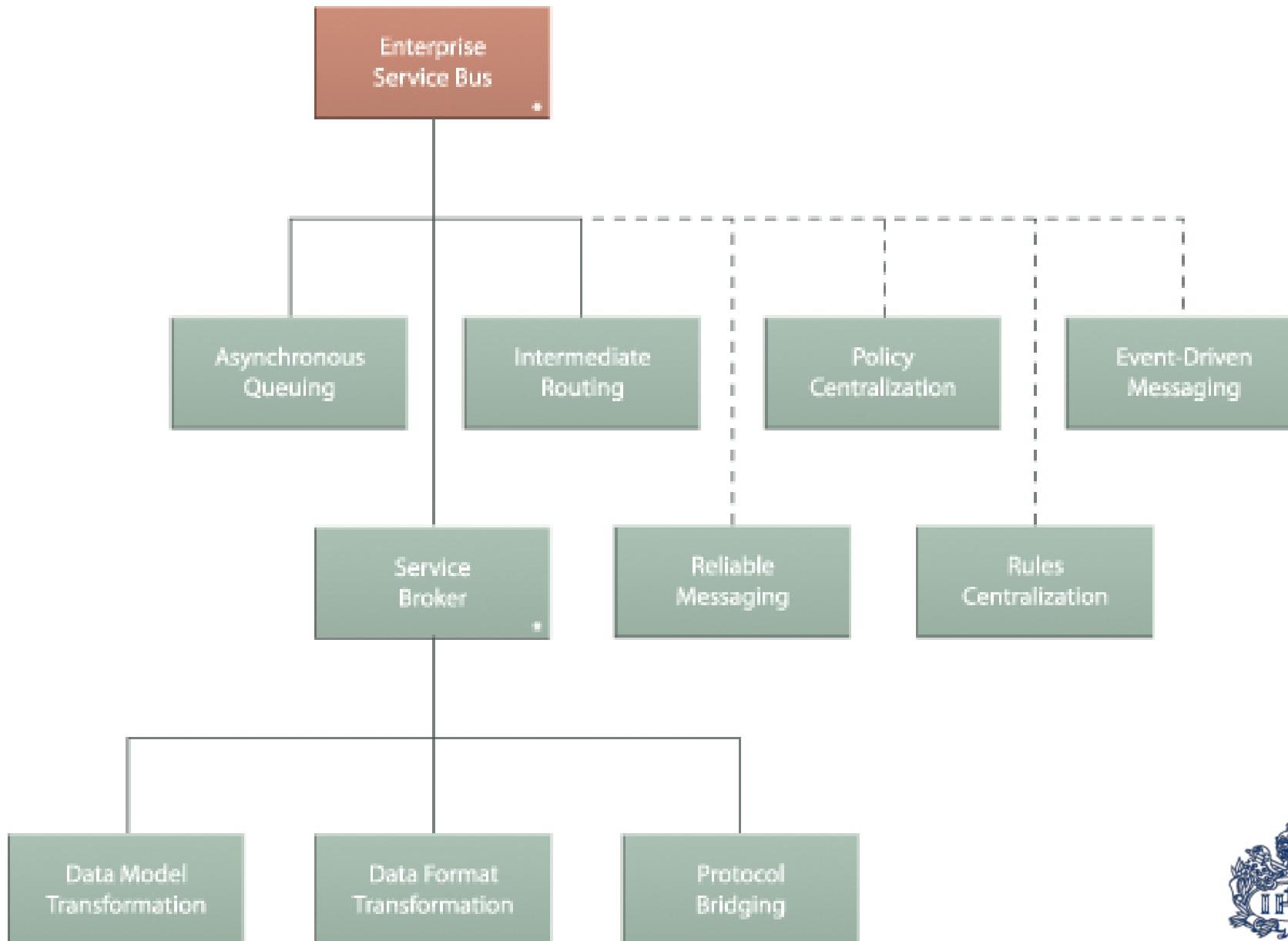
Three-Layer
Inventory
Uniform Contract



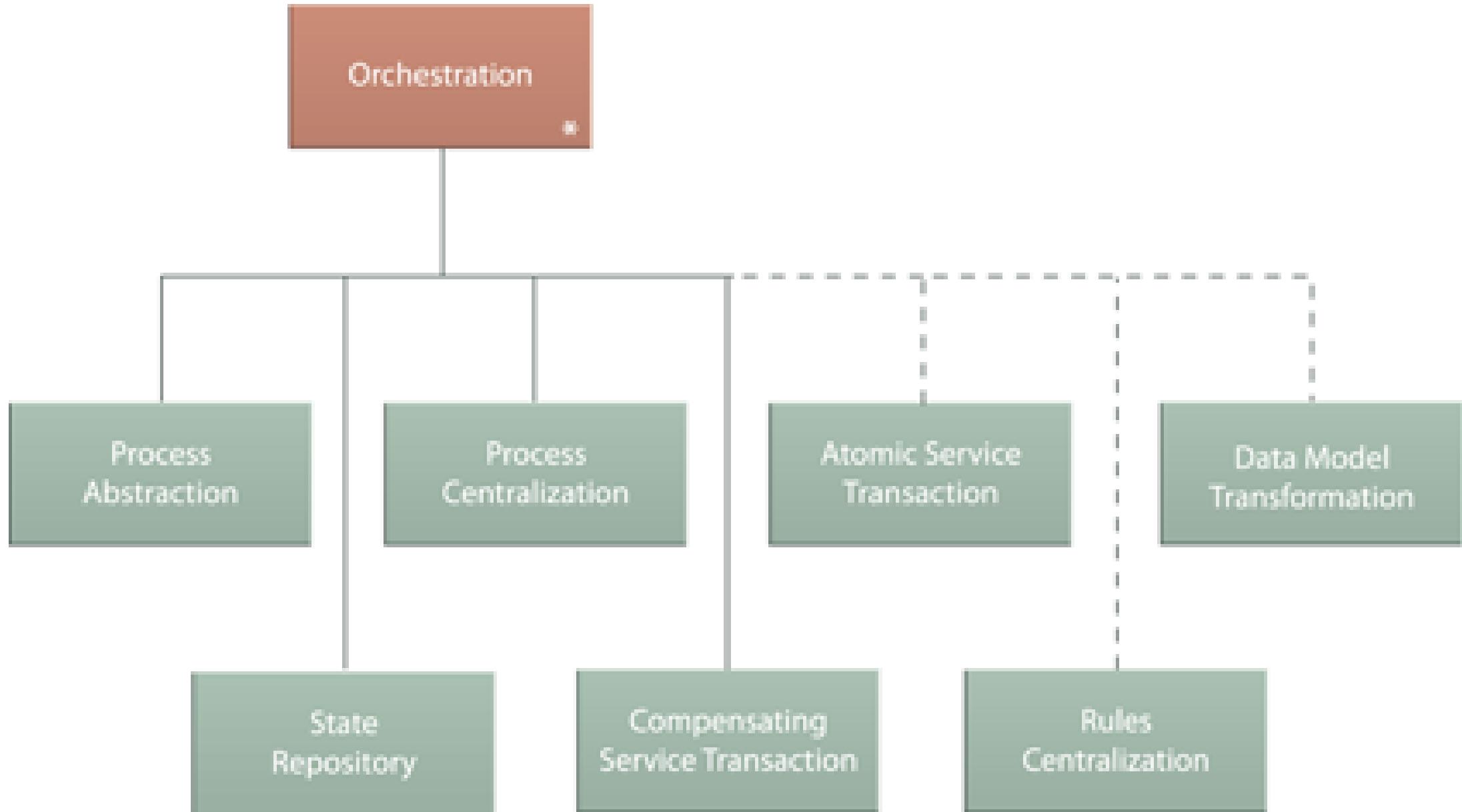
Patrones Diseño SOA Official Endpoint



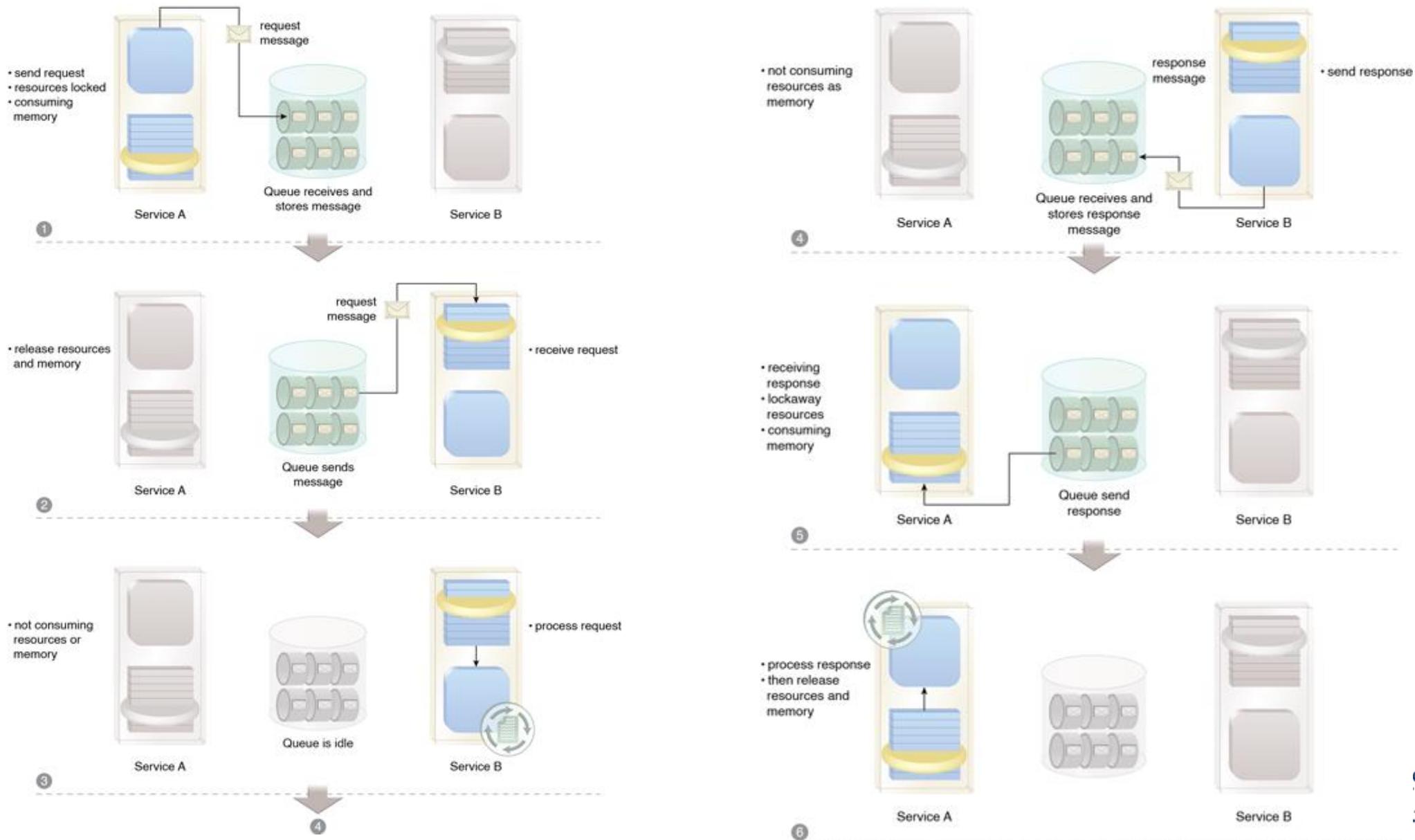
Patrones Diseño SOA Enterprise Service Bus



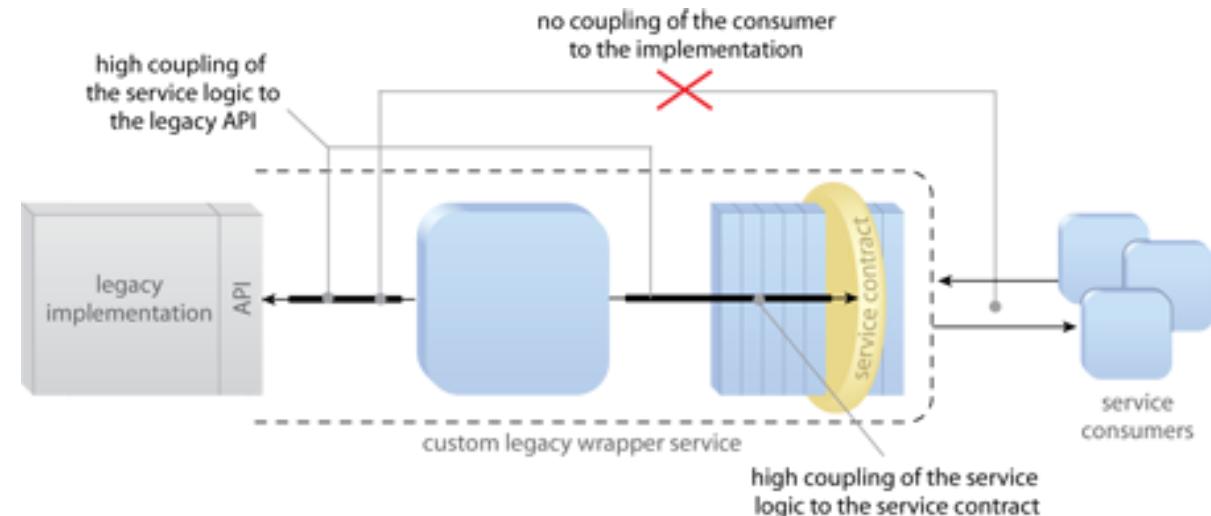
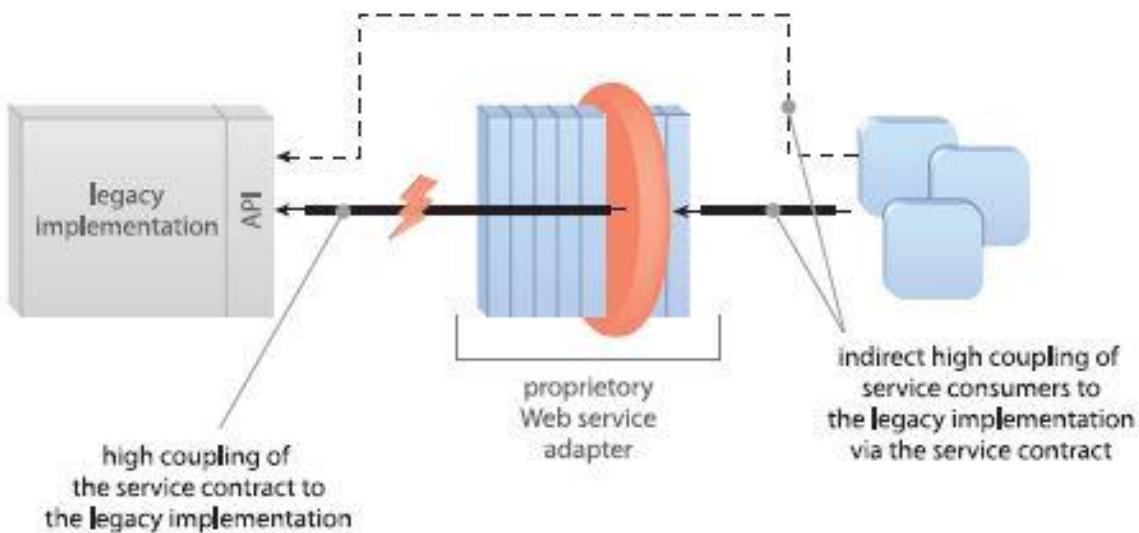
Patrones Diseño SOA Orchestration



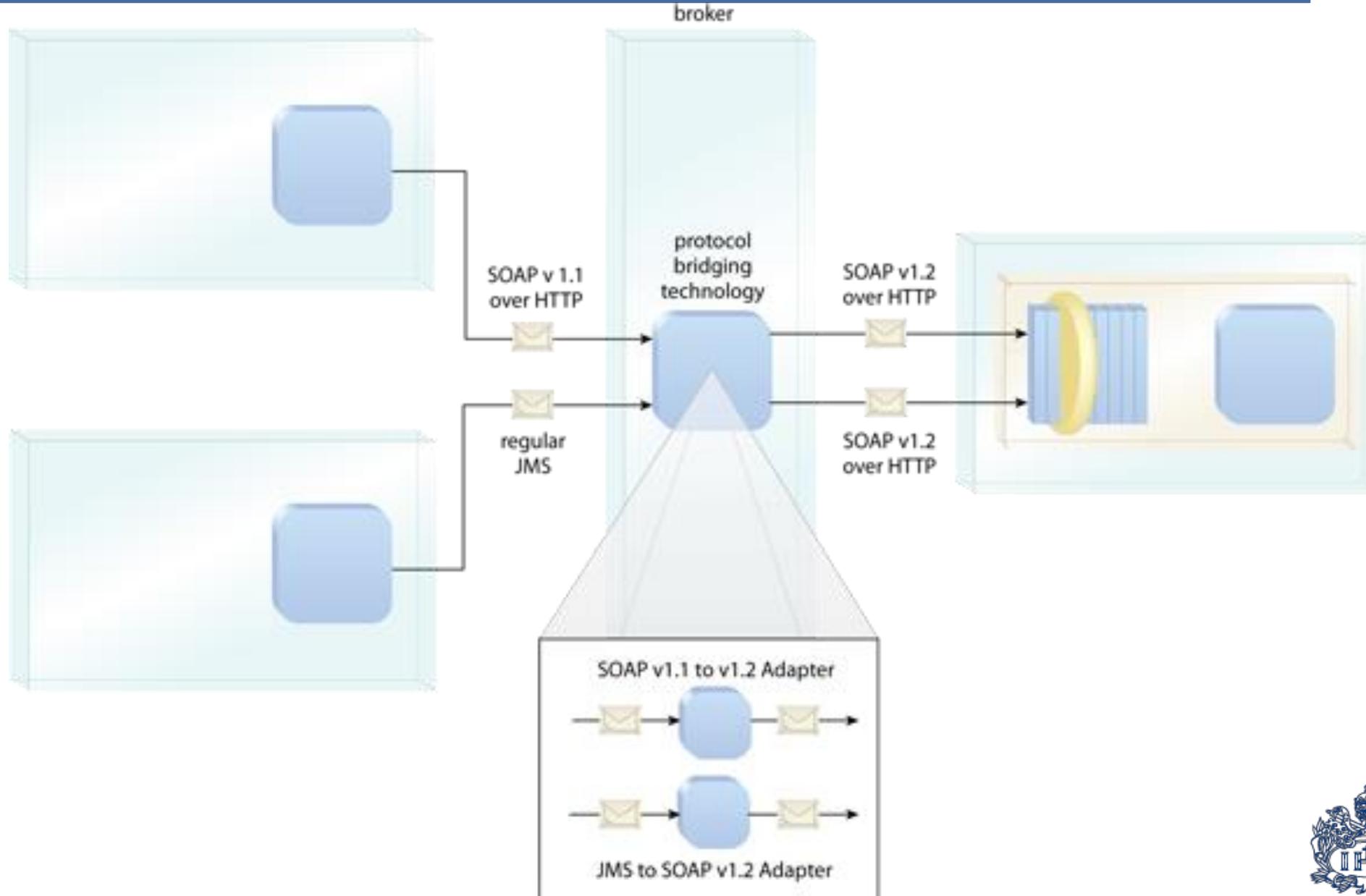
Patrones Diseño SOA - Asynchronous Queuing



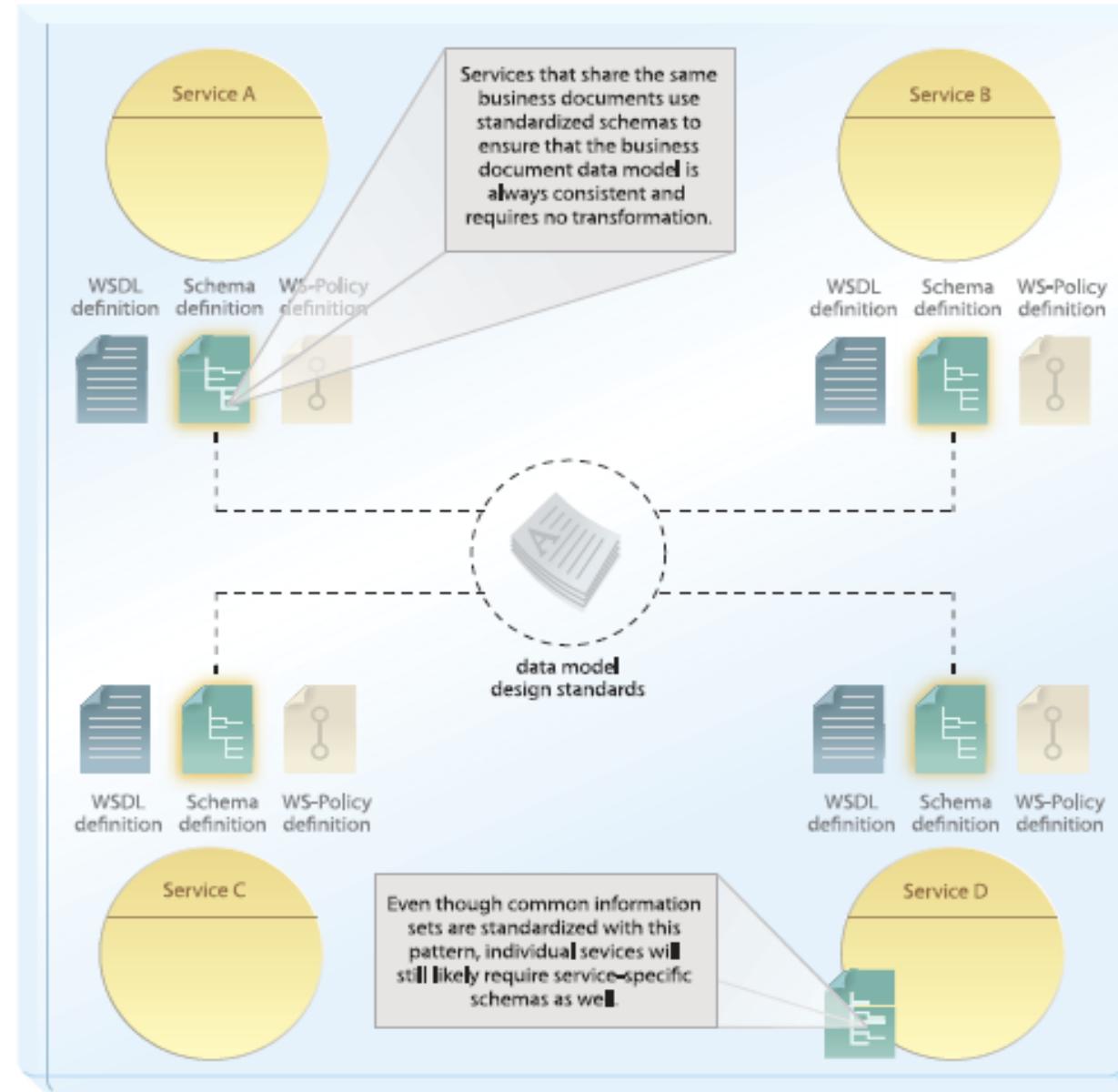
Patrones Diseño SOA - Legacy Wrapper



Patrones Diseño SOA - Protocol Bridging

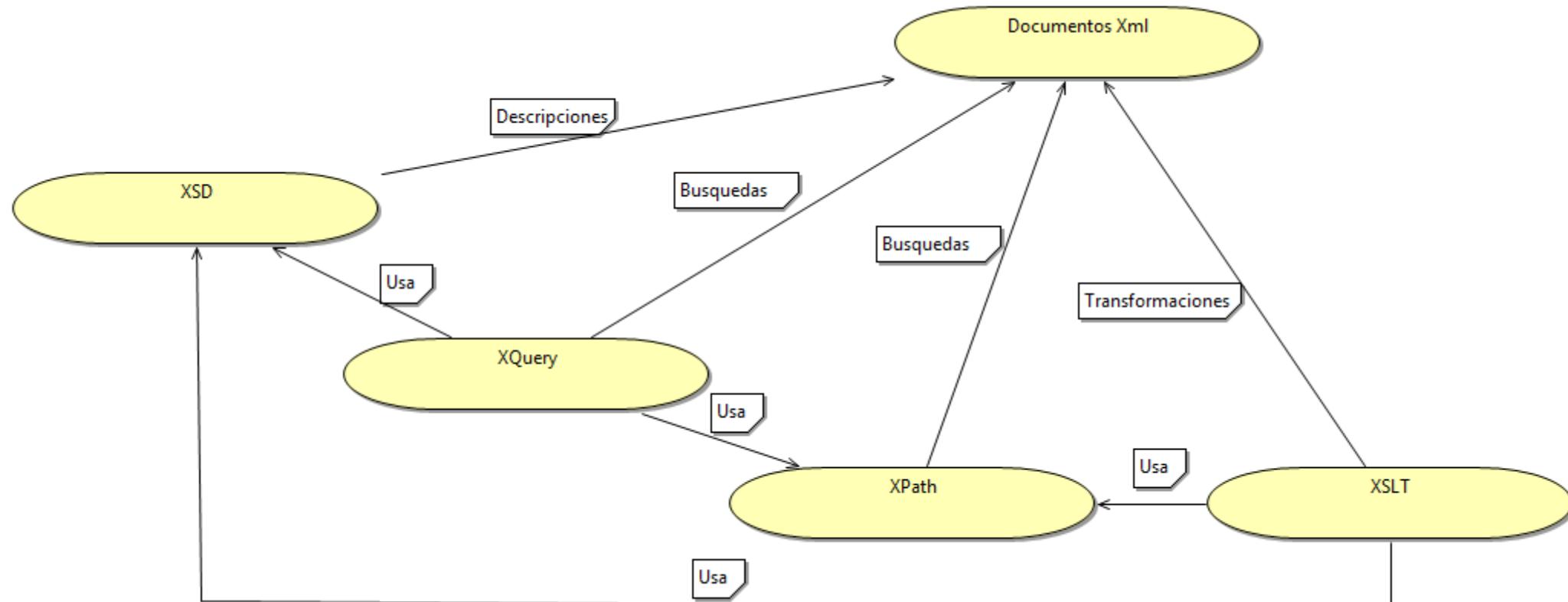


Patrones Diseño SOA - Canonical Schema

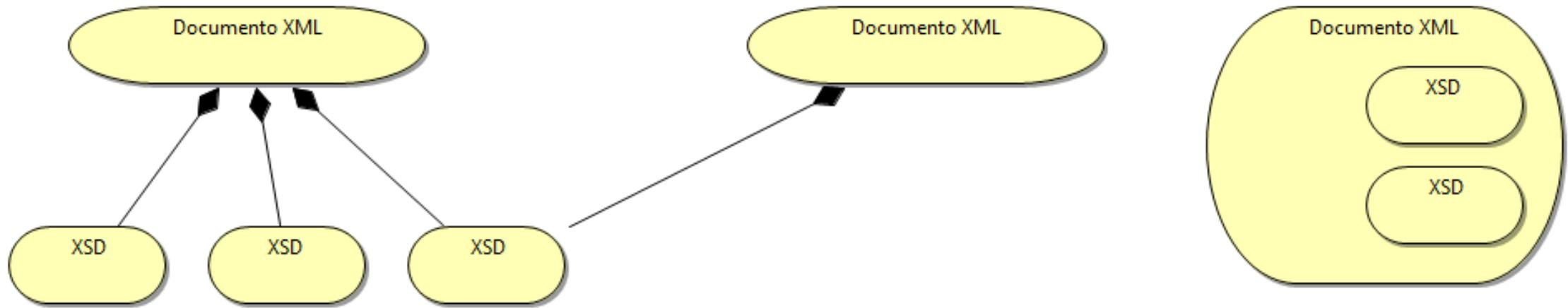


Generalidades de los Web Services

Documento XML



Definiciones Esquemas XML



Tipos de Esquemas XML

- Los elementos en los esquemas XML que contienen elementos hijos o atributos que son tipos complejos (**complex types.**)

- Los elementos que solo contienen valores son tipos simples.

```
<element name="orderId"
        type="string"
        nillable="true"/>
```

- Todos los atributos tienen tipos simples porque solo contienen valores.

- Los tipos simples (Simple types) son usados para incorporar tipos de datos (string, int entre otros).

- Tipos complejo, son tipos construidos para hacer mini jerarquías de elementos que pueden representar cuerpo complejos con de datos relacionados

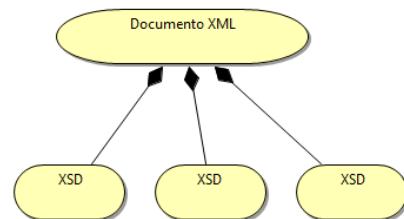
- Cuando se crea el contrato de un servicio, es común representar tipos complejos en toda la estructura de mensajes de entrada y salida.

```
<element name="orderQuouteElement" >
    <complexType>
        <sequence>
            <element name="orderId" type="string" nillable="true"/>
        </sequence>
    </complexType>
</element>
```



Namespaces

- Los espacio de nombre o **NAMESPACES** son una importante parte de Lenguaje basado en XML que establece la medio de **crear una única identificación**.
- Todos los lenguajes XML (XSD, WSDL , SOAP , WS-Adresing, SW-BPEL) tiene un **NAMESPACES** predefinido asociado con el.
- Donde los elementos XML se mezclan juntos en un mismo XML en tiempo de ejecución se debe averiguar de que lenguaje pertenece cada uno.



- Una declaración de un **NAMESPACES** establece un prefijo (**prefix**)

```
xmlns:xsd1="http://sony.com.co/orderQuoute"

<message name="SonyOrderService_orderQuoute">
    <part name="parameters" element="xsd1:orderQuouteElement"/>
</message>
```

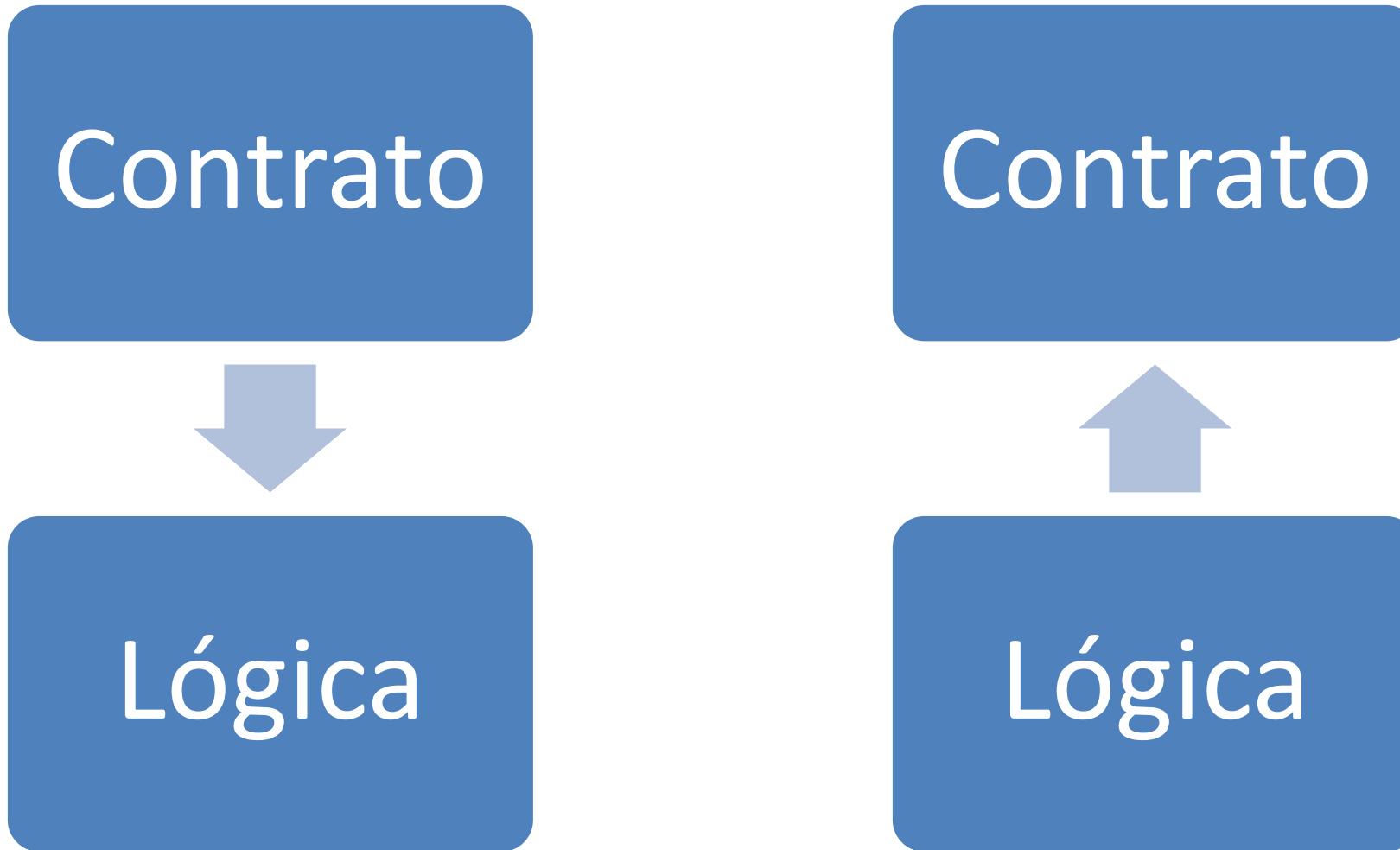
Contrato servicio WEB



- **portType:** contiene conjunto de operaciones abstractas que representan los tipos de correspondencia que pueden producirse entre el cliente y el servidor.
- **operation:** define la operación.
- **message:** define los datos que participan en una operación.
- **binding:** especificamos los protocolos de comunicación.
- **port:** definen la dirección de un binding.
- **service:** conjunto de puertos.

Taller WebServices JAX-WS

Top-Down - Bottom-up



WSDL Orden Cotización

Comunicación con SONY (WSDL)

Ordenar Cotización

```
<!-- Ordenar Cotizacion -->
<types>
    <schema>
        <element name="orderQuouteElement">
            <complexType>
                <sequence>
                    <element name="orderId" type="string" nillable="true"/>
                </sequence>
            </complexType>
        <element name="orderQuouteResponseElement">
            <complexType>
                <sequence>
                    <element name="result" type="boolean"/>
                </sequence>
            </complexType>
        </element>
    </schema>
</types>
<message name="SonyOrderService_orderQuoute">
    <part name="parameters" element="tns0:orderQuouteElement"/>
</message>
<message name="SonyOrderService_orderQuouteResponse">
    <part name="parameters" element="tns0:orderQuouteResponseElement"/>
</message>
<portType name="SonyOrderService">
    <operation name="orderQuoute">
        <input message="tns:SonyOrderService_orderQuoute"/>
        <output message="tns:SonyOrderService_orderQuouteResponse"/>
    </operation>
</portType>
```



```
        <!-- Ordenar Cotizacion -->
<types>
    <schema>
        <elementname="orderQuouteElement">
            <complexType>
                <sequence>
                    <elementname="orderId" type="string" nillable="true"/>
                </sequence>
            </complexType>
        <elementname="orderQuouteResponseElement">
            <complexType>
                <sequence>
                    <elementname="result" type="boolean"/>
                </sequence>
            </complexType>
        </element>
    </schema>
</types>
<messagename="SonyOrderService_orderQuoute">
    <partname="parameters" element="tns0:orderQuouteElement"/>
</message>
<messagename="SonyOrderService_orderQuouteResponse">
    <partname="parameters" element="tns0:orderQuouteResponseElement"/>
</message>
<portTypename="SonyOrderService">
    <operationname="orderQuoute">
        <inputmessage="tns:SonyOrderService_orderQuoute"/>
        <outputmessage="tns:SonyOrderService_orderQuouteResponse"/>
    </operation>
</portType>
```



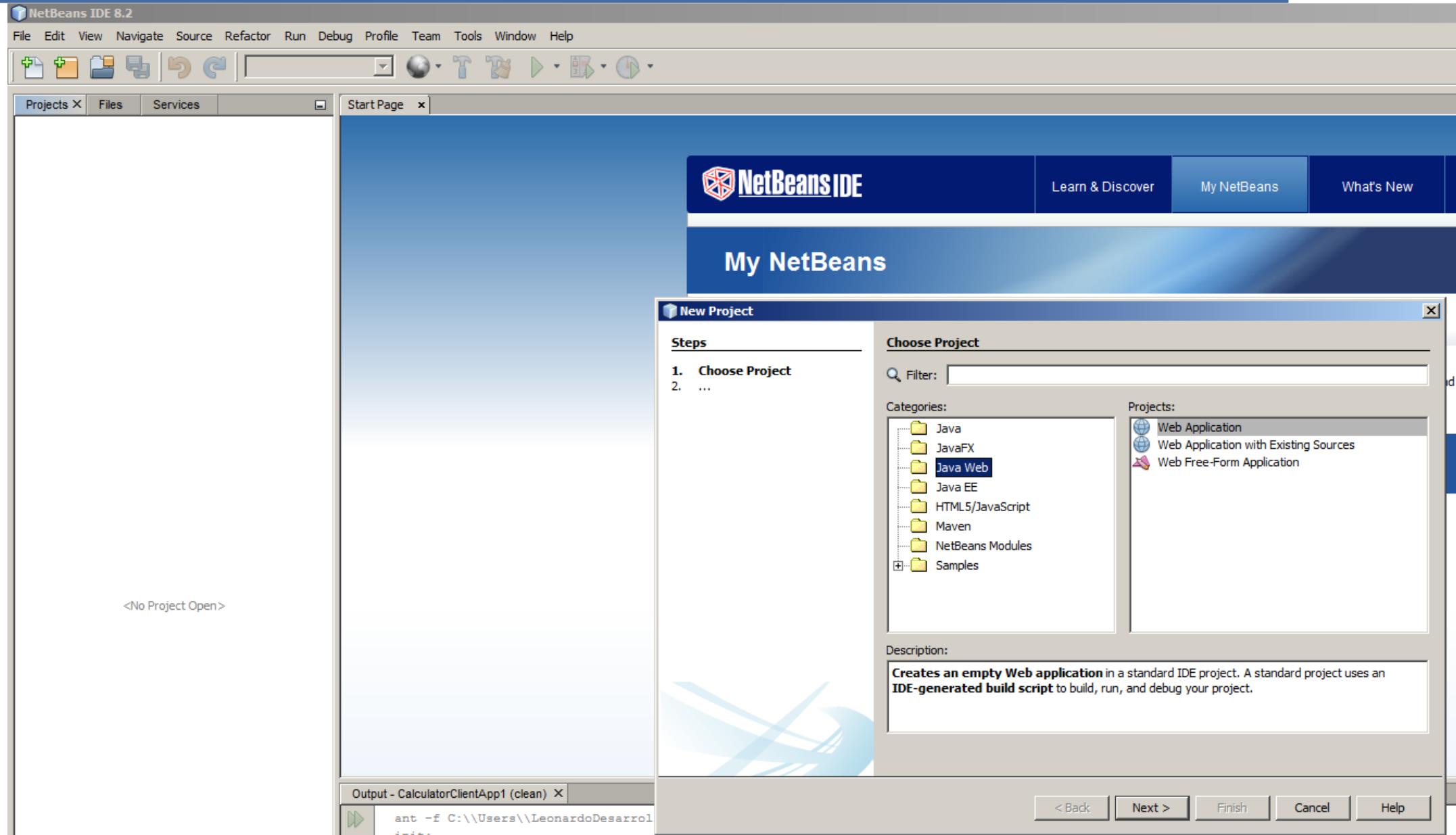
WSDL Orden Cotización Arreglos



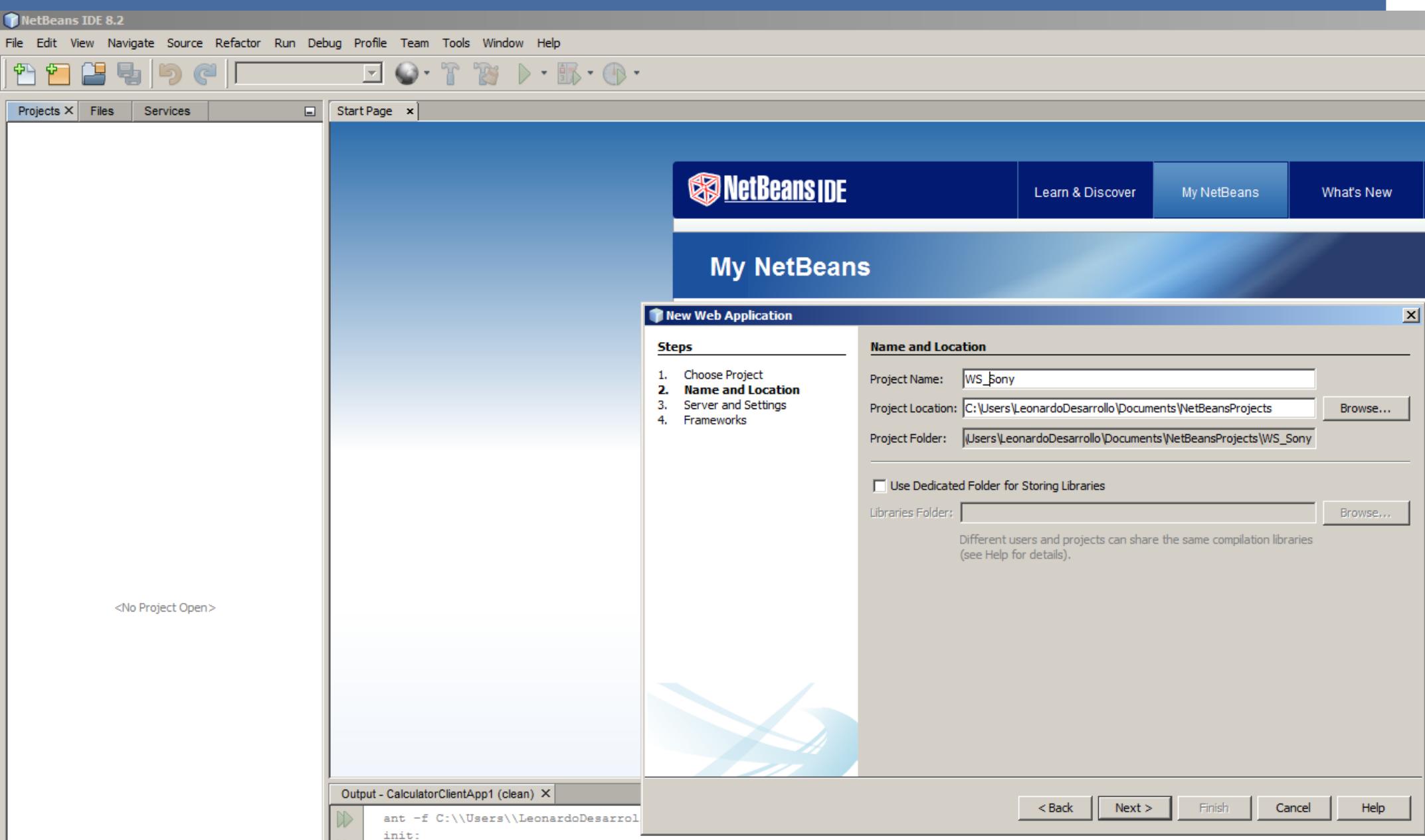
SonyService.wsdl

\PICA segundo semestre 2017 documentos compartidos\Presentaciones\Files\SonyService.wsdl

Nuevo Proyecto (Web Application)



Nombre Proyecto WS-Sony



Pontificia Universidad
JAVIERIANA
Bogotá

Configuración del servidor

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects X Files Services Start Page x

<No Project Open>

New Web Application

Steps

1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

Server: GlassFish Server 4.1.1 Add...

Java EE Version: Java EE 7 Web

Context Path: /WS_Sony

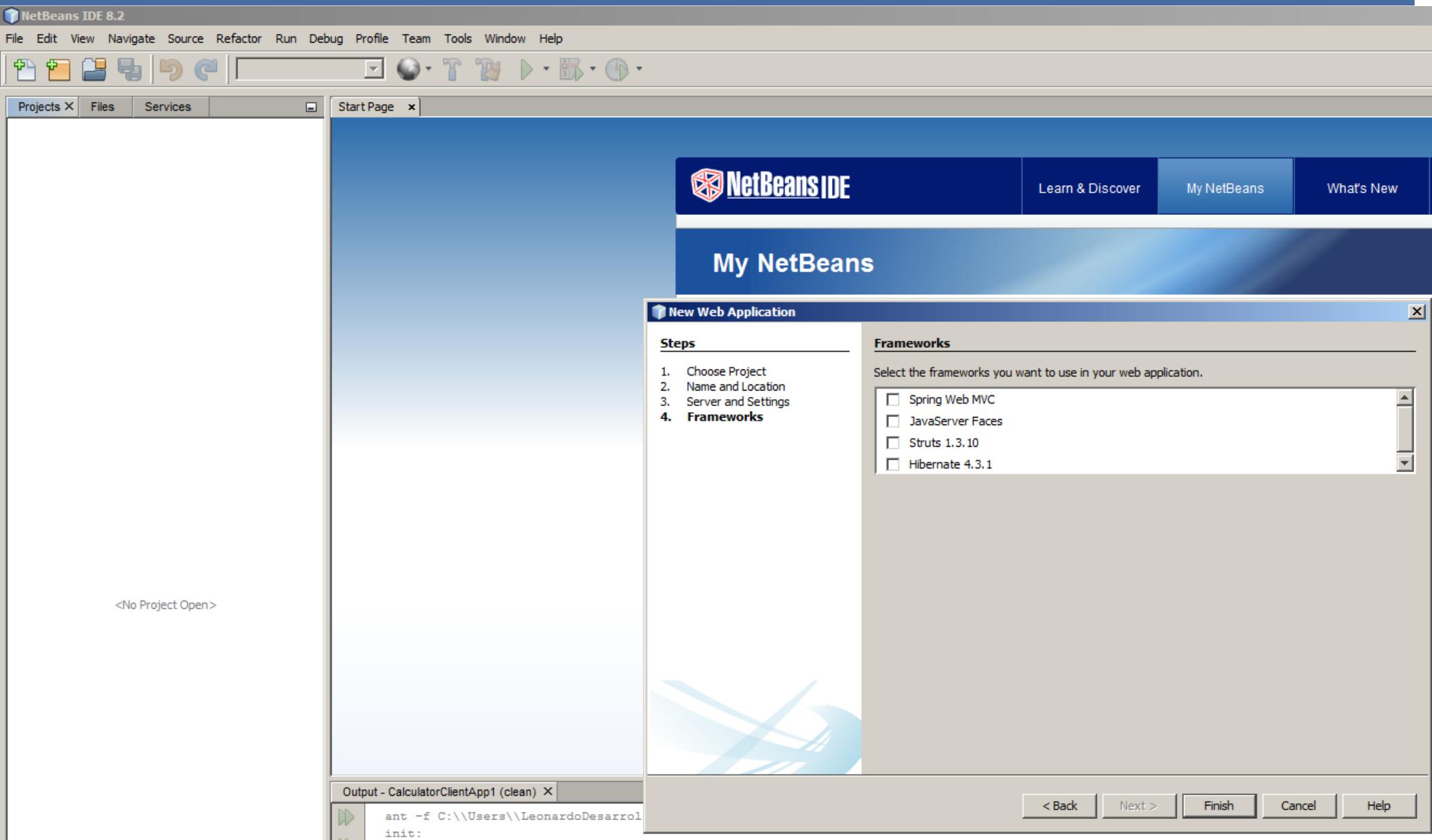
Output - CalculatorClientApp1 (clean) x

```
ant -f C:\\\\Users\\\\LeonardoDesarrol  
init:
```

< Back Next > Finish Cancel Help



Sin Ningún Framework



Pontificia Universidad
JAVIERIANA
Bogotá

Nuevo Paquete co.com.sony

WS_Sony - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects X Files Services

WS_Sony

- Web Pages
- Source Packages
 - <default package>
- Libraries
- Enterprise Beans
- Configuration Files

Start Page x index.html x

Source History

```
1 <!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project Properties.
4 To change this template file, choose Tools | Templates
5 and open the template in the editor.
6 -->
7 <html>
8   <head>
9     <title>TODO supply a
10    <meta charset="UTF-8"
11    <meta name="viewport"
12    </head>
13   <body>
14     <div>TODO write conte
15   </body>
16 </html>
17
```

New Java Package

Steps

- Choose File Type
- Name and Location

Name and Location

Package Name: co.com.sony

Project: WS_Sony

Location: Source Packages

Created Folder: rs\LeonardoDesarrollo\Documents\NetBeansProjects\WS_Sony\src\java\co\com\sony

Navigator X

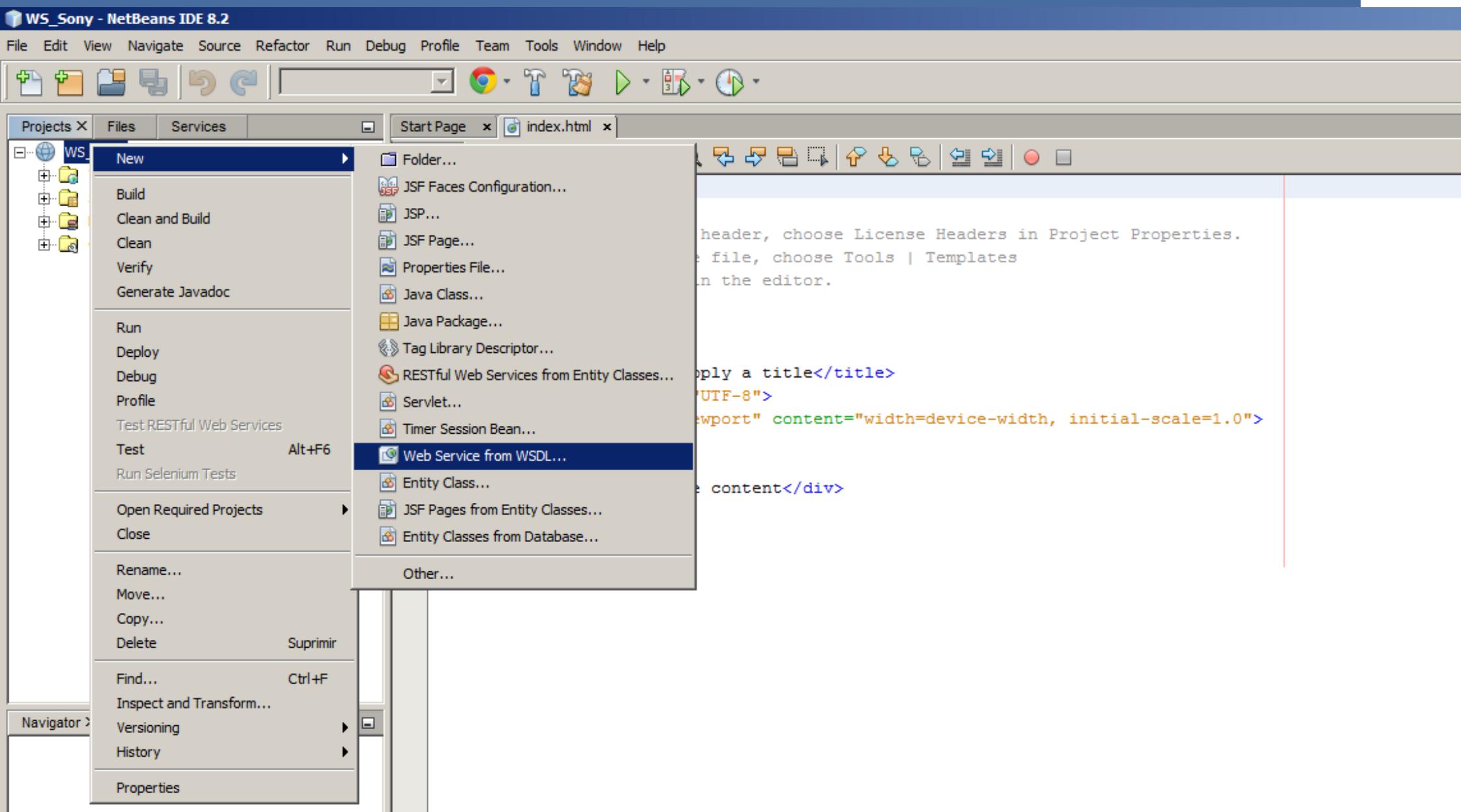
+ html

Output X

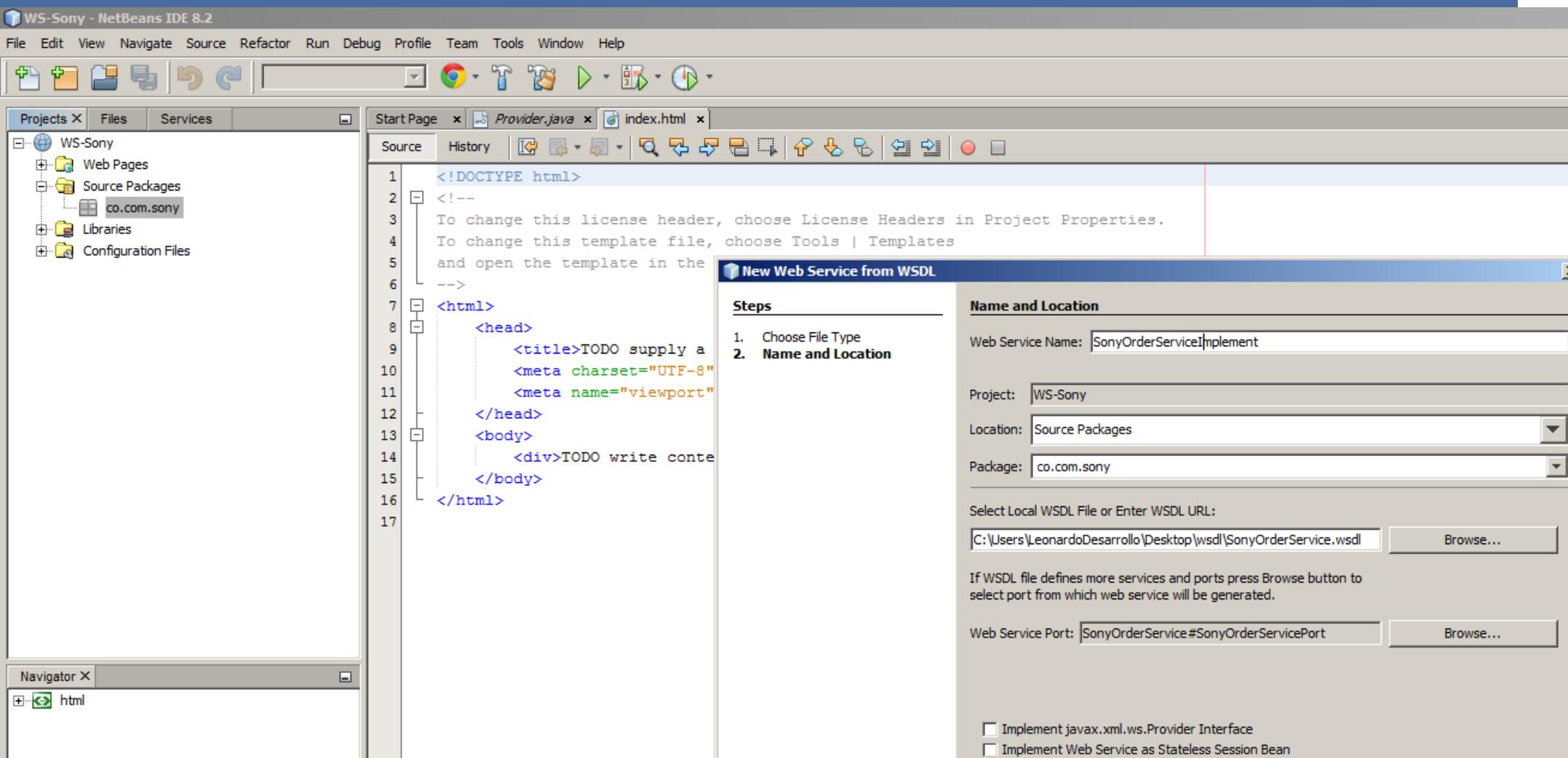
XML check x Retriever Output x Java DB Database Process x Glassfish Server 4.1.1 x TWS SURV (team) x



Nuevo Servicio desde un WSDL



Nombramos el servicio y seleccionamos el archivo WSDL



Se genera el código para implementar el WS

The screenshot shows the NetBeans IDE 8.2 interface with the title bar "WS-Sony - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar has various icons for file operations like New, Open, Save, and Cut/Paste. The Projects tab shows a project named "WS-Sony" with subfolders like "Web Pages", "Source Packages", and "Generated Sources (jax-ws)". The "Generated Sources (jax-ws)" folder contains files such as "SonyOrderService.java", "SonyOrderServicePortType.java", "ObjectFactory.java", "OrderQuouteElement.java", "OrderQuouteResponseElement.java", and "package-info.java". The main editor window displays the source code for "SonyOrderServiceImplement.java". The code is annotated with comments and annotations:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package co.com.sony;

import javax.jws.WebService;

/**
 *
 * @author LeonardoDesarrollo
 */
@WebService(serviceName = "SonyOrderService", portName = "SonyOrderServicePort", endpointInterface = "co.com.sony.SonyOrderServicePortType", targetNamespace = "http://sony.com.co/", wsdlLoca
public class SonyOrderServiceImplement {

    public co.com.sony.orderquoute.OrderQuouteResponseElement orderQuoute(co.com.sony.orderquoute.OrderQuouteElement parameters) {
        //TODO implement this method
        throw new UnsupportedOperationException("Not implemented yet.");
    }
}
```

The "Output" tab at the bottom shows "Retriever Output" and "WS-Sony (wsimport-service-SonyOrderServiceImplement)".

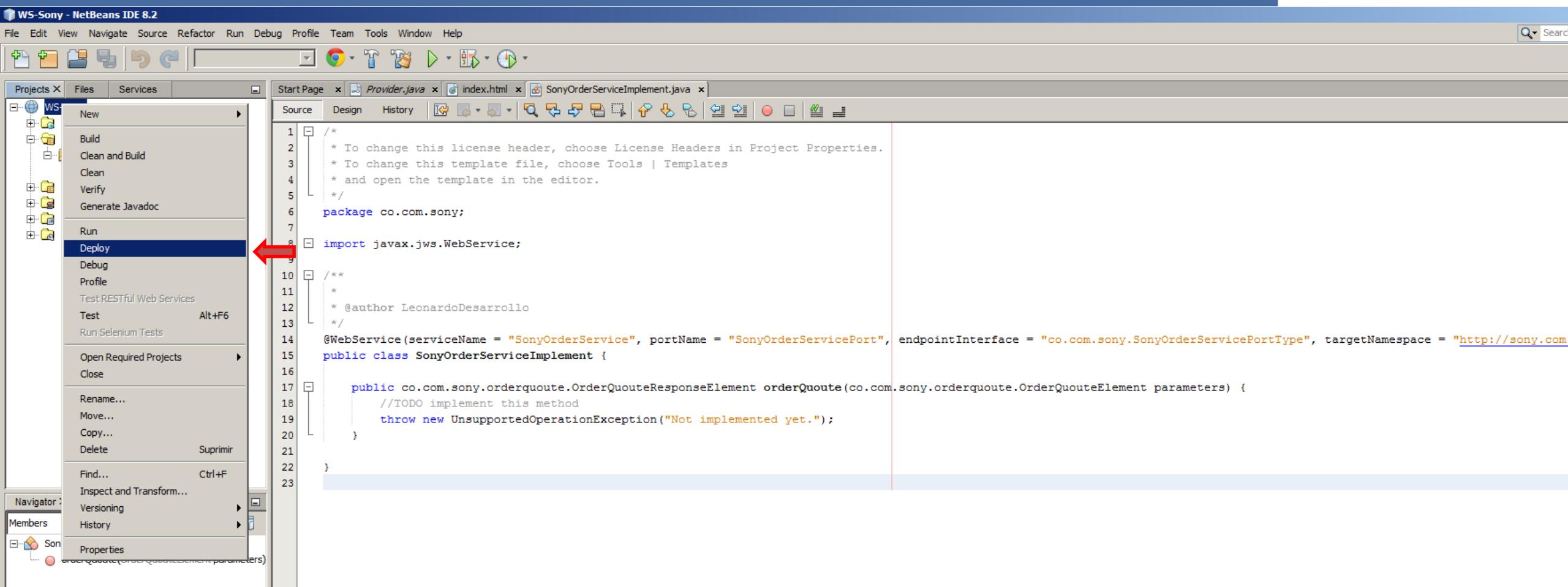
Se genera el código para implementar el WS

The screenshot shows the NetBeans IDE 8.2 interface with the title bar "WS-Sony - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar has various icons for file operations like New, Open, Save, and Cut/Paste. The Projects X tab is selected in the left sidebar, showing a project named "WS-Sony" with subfolders like "Web Pages" (containing "WEB-INF" and "index.html"), "Source Packages" (containing "co.com.sony" with "SonyOrderServiceImplement.java"), and "Generated Sources (jax-ws)" (containing "co.com.sony" with "SonyOrderService.java", "SonyOrderServicePortType.java", "co.com.sony.orderquoute" with "ObjectFactory.java", "OrderQuouteElement.java", "OrderQuouteResponseElement.java", and "package-info.java"). The main editor window displays the "SonyOrderServiceImplement.java" source code:

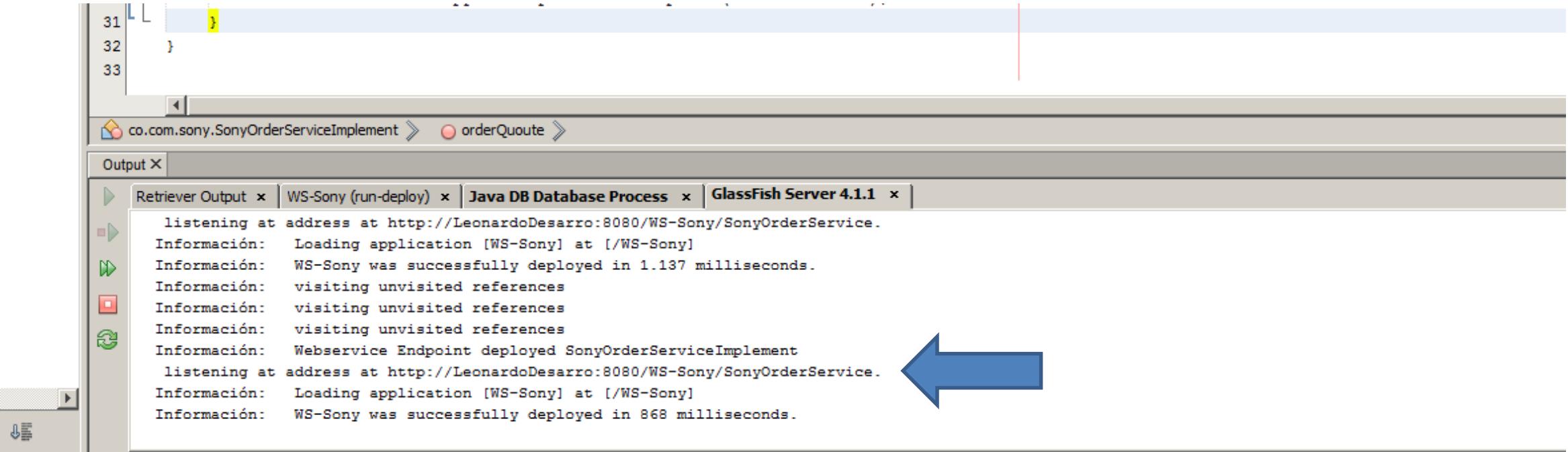
```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package co.com.sony;
7
8 import javax.jws.WebService;
9
10 /**
11 *
12 * @author LeonardoDesarrollo
13 */
14 @WebService(serviceName = "SonyOrderService", portName = "SonyOrderServicePort", endpointInterface = "co.com.sony.SonyOrderServicePortType", targetNamespace = "http://sony.com.co/", wsdlLoca
15 public class SonyOrderServiceImplement {
16
17     public co.com.sony.orderquoute.OrderQuouteResponseElement orderQuoute(co.com.sony.orderquoute.OrderQuouteElement parameters) {
18         //TODO implement this method
19         throw new UnsupportedOperationException("Not implemented yet.");
20     }
21
22 }
23
```

The code editor has a toolbar above it with various icons for navigation and code manipulation. The bottom of the screen shows the Navigator X and Output X panes.

Desplegamos el proyecto WS



Desplegamos el proyecto WS



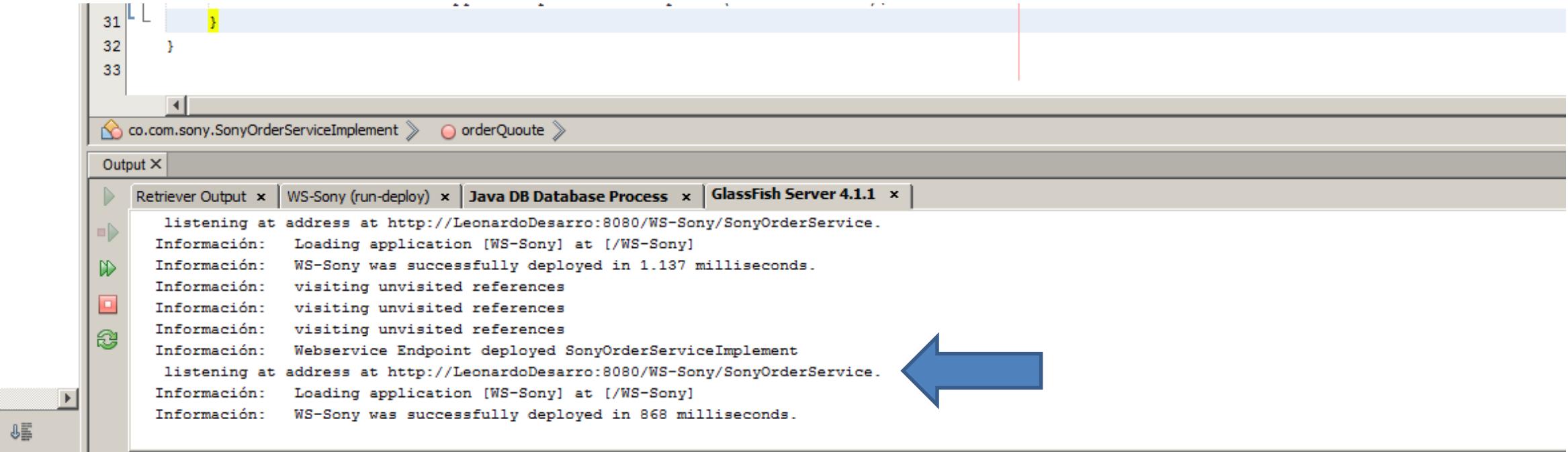
The screenshot shows an IDE interface with the following details:

- Code Editor:** Shows lines 31, 32, and 33 of a Java file.
- Project Explorer:** Displays two projects: "co.com.sony.SonyOrderServiceImplement" and "orderQuoute".
- Output Window:** Titled "GlassFish Server 4.1.1", it contains the following deployment log:

```
listening at address at http://LeonardoDesarro:8080/WS-Sony/SonyOrderService.
Información: Loading application [WS-Sony] at [/WS-Sony]
Información: WS-Sony was successfully deployed in 1.137 milliseconds.
Información: visiting unvisited references
Información: visiting unvisited references
Información: visiting unvisited references
Información: Webservice Endpoint deployed SonyOrderServiceImplement
listening at address at http://LeonardoDesarro:8080/WS-Sony/SonyOrderService.
Información: Loading application [WS-Sony] at [/WS-Sony]
Información: WS-Sony was successfully deployed in 868 milliseconds.
```

A large blue arrow points from the right side of the slide towards the "GlassFish Server 4.1.1" tab in the output window.

Desplegamos el proyecto WS



The screenshot shows an IDE interface with the following details:

- Code Editor:** Shows lines 31, 32, and 33 of a Java file.
- Project Explorer:** Displays two projects: "co.com.sony.SonyOrderServiceImplement" and "orderQuoute".
- Output Window:** Titled "GlassFish Server 4.1.1", it contains the following deployment log:

```
listening at address at http://LeonardoDesarro:8080/WS-Sony/SonyOrderService.
Información: Loading application [WS-Sony] at [/WS-Sony]
Información: WS-Sony was successfully deployed in 1.137 milliseconds.
Información: visiting unvisited references
Información: visiting unvisited references
Información: visiting unvisited references
Información: Webservice Endpoint deployed SonyOrderServiceImplement
listening at address at http://LeonardoDesarro:8080/WS-Sony/SonyOrderService.
Información: Loading application [WS-Sony] at [/WS-Sony]
Información: WS-Sony was successfully deployed in 868 milliseconds.
```

A large blue arrow points from the right side of the slide towards the "GlassFish Server 4.1.1" output window.

Validamos el despliegue del WS

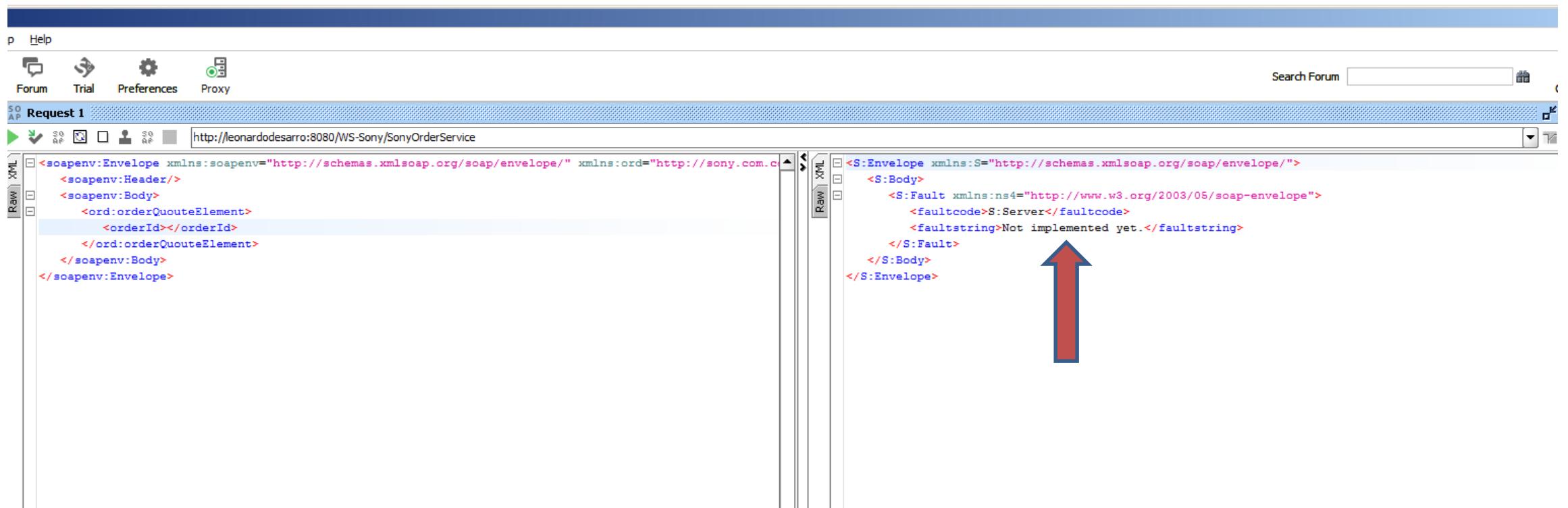


A screenshot of a web browser window titled "Servicios Web". The address bar contains the URL "leonardodesarro:8080/WS-Sony/SonyOrderService". A blue arrow points from the left towards the URL bar.

Servicios web

| Punto Final | Información |
|---|---|
| Nombre de Servicio\ : {http://sony.com.co/}SonyOrderService | Dirección\ : http://leonardodesarro:8080/WS-Sony/SonyOrderService |
| Nombre de Puerto\ : {http://sony.com.co/}SonyOrderServicePort | WSDL\ : http://leonardodesarro:8080/WS-Sony/SonyOrderService?wsdl |
| | Clase de Implementación\ : co.com.sony.SonyOrderServiceImplement |

Probamos el despliegue del WS



The screenshot shows a SOAP UI interface with the following details:

Request 1 (Left Panel):
URL: <http://leonardodesarro:8080/WS-Sony/SonyOrderService>
Raw XML Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ord="http://sony.com.c...</soapenv:Envelope>
```

Response 1 (Right Panel):
Raw XML Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope"><faultcode>S:Server</faultcode><faultstring>Not implemented yet.</faultstring></S:Fault></S:Body></S:Envelope>
```

A large red arrow points upwards from the bottom of the response XML towards the `<faultstring>Not implemented yet.</faultstring>` element.



Cambiamos la implementación del WS

```
Source Design History |  |        |     |  
```

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6 package co.com.sony;
7
8 import co.com.sony.orderquoute.OrderQuouteResponseElement;
9 import javax.jws.WebService;
10
11 /**
12 *
13 * @author LeonardoDesarrollo
14 */
15 @WebService(serviceName = "SonyOrderService", portName = "SonyOrderServicePort", endpointInterface = "co.com.sony.SonyOrderServicePortType", targetNamespace = "http://sony.com.co/", wsdlLoca
16 public class SonyOrderServiceImplement {
17
18     public co.com.sony.orderquoute.OrderQuouteResponseElement orderQuoute(co.com.sony.orderquoute.OrderQuouteElement parameters) {
19         //TODO implement this method
20         if(parameters!=null)
21         {
22             if (parameters.getOrderId()!=null)
23             {
24                 co.com.sony.orderquoute.OrderQuouteResponseElement orderQuouteResponseElement = new OrderQuouteResponseElement();
25                 orderQuouteResponseElement.setResult(true);
26                 return orderQuouteResponseElement;
27             }
28             else throw new UnsupportedOperationException("Error nu llego la orden ");
29         }
30         else throw new UnsupportedOperationException("Error Null ");
31     }
32 }
```



Probamos el WS

The screenshot shows a SOAP UI interface with the following details:

- Toolbar:** Help, Forum, Trial, Preferences, Proxy, Search Forum.
- Request 1:** URL: <http://leonardodesarro:8080/WS-Sony/SonyOrderService>
- Raw Request XML:**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ord="http://sony.com.co/orderQuoute">
<soapenv:Header/>
<soapenv:Body>
<ord:orderQuouteElement>
<orderId>123</orderId>
</ord:orderQuouteElement>
</soapenv:Body>
</soapenv:Envelope>
```
- Raw Response XML:**

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:orderQuouteResponseElement xmlns:ns2="http://sony.com.co/orderQuoute">
<result>true</result>
</ns2:orderQuouteResponseElement>
</S:Body>
</S:Envelope>
```

Probamos el WS forzando un error

The screenshot shows the SoapUI interface with a request and response pane. The request pane contains a SOAP envelope with an order quote element. The response pane shows a fault message with a fault string 'Error no llego la orden'. A red arrow points from the request pane to the response pane.

Request 1

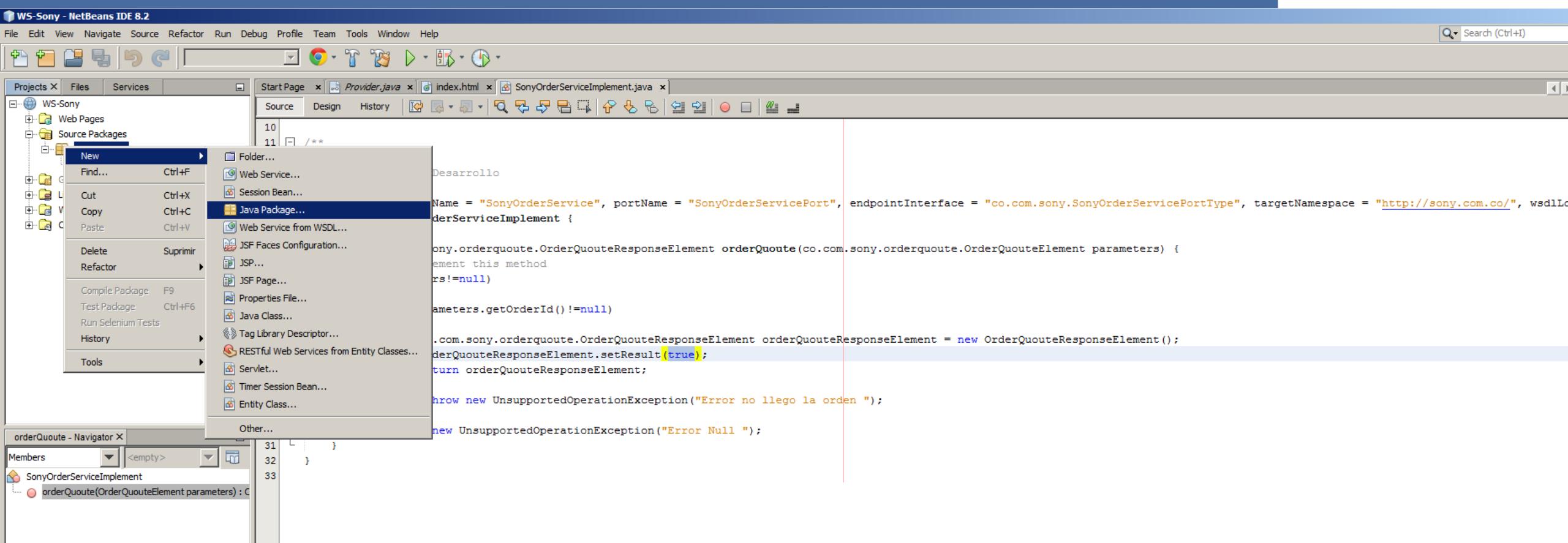
http://leonardodesarro:8080/WS-Sony/SonyOrderService

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ord="http://sony.com.cu">
  <soapenv:Header/>
  <soapenv:Body>
    <ord:orderQuouteElement>
      <ord:orderQuouteElement>
        </ord:orderQuouteElement>
      </ord:orderQuouteElement>
    </soapenv:Body>
  </soapenv:Envelope>
```

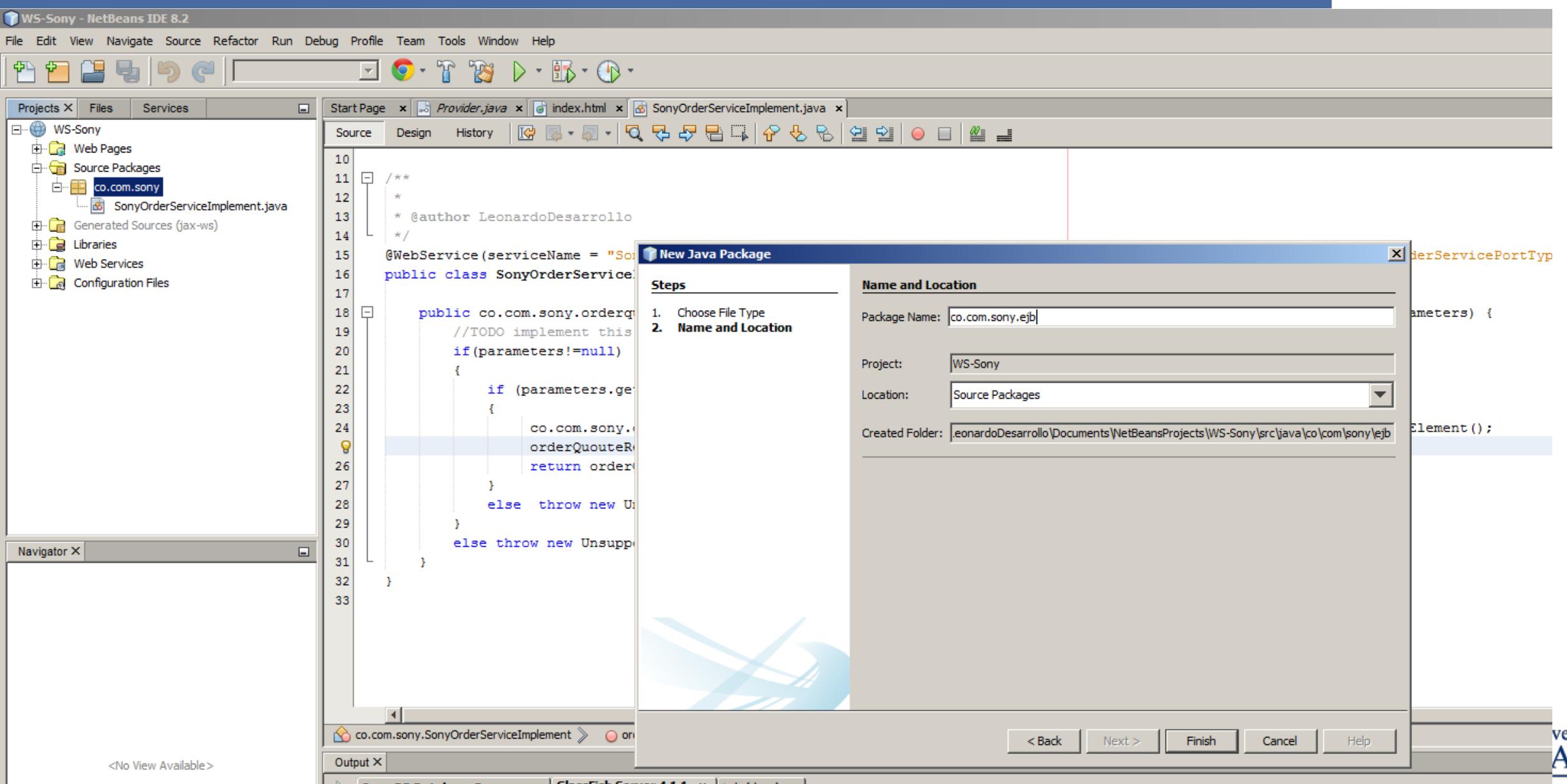
```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>S:Server</faultcode>
      <faultstring>Error no llego la orden</faultstring>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

JAX-WS Bottom-Up Proyecto Sony

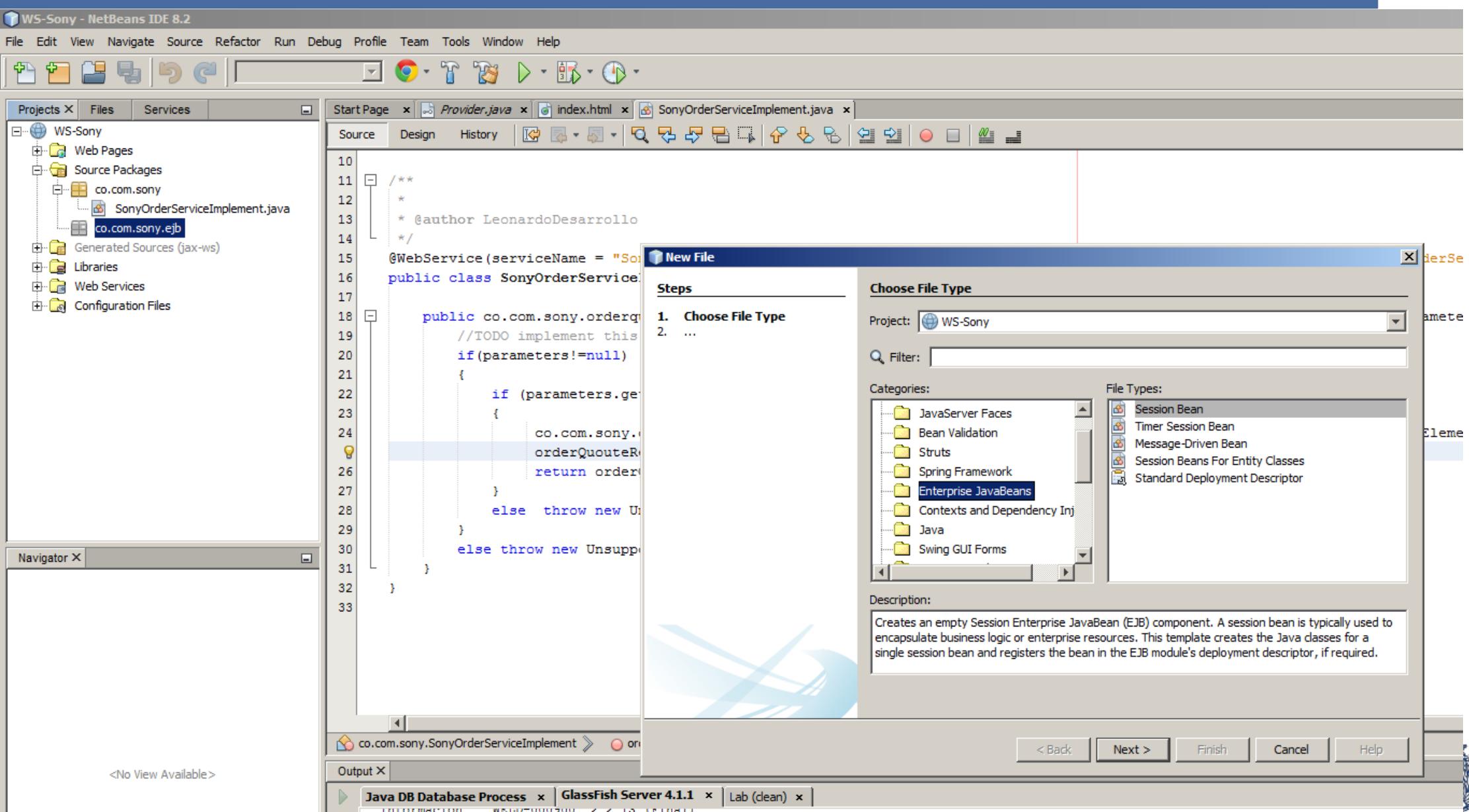
Creamos un nuevo paquete



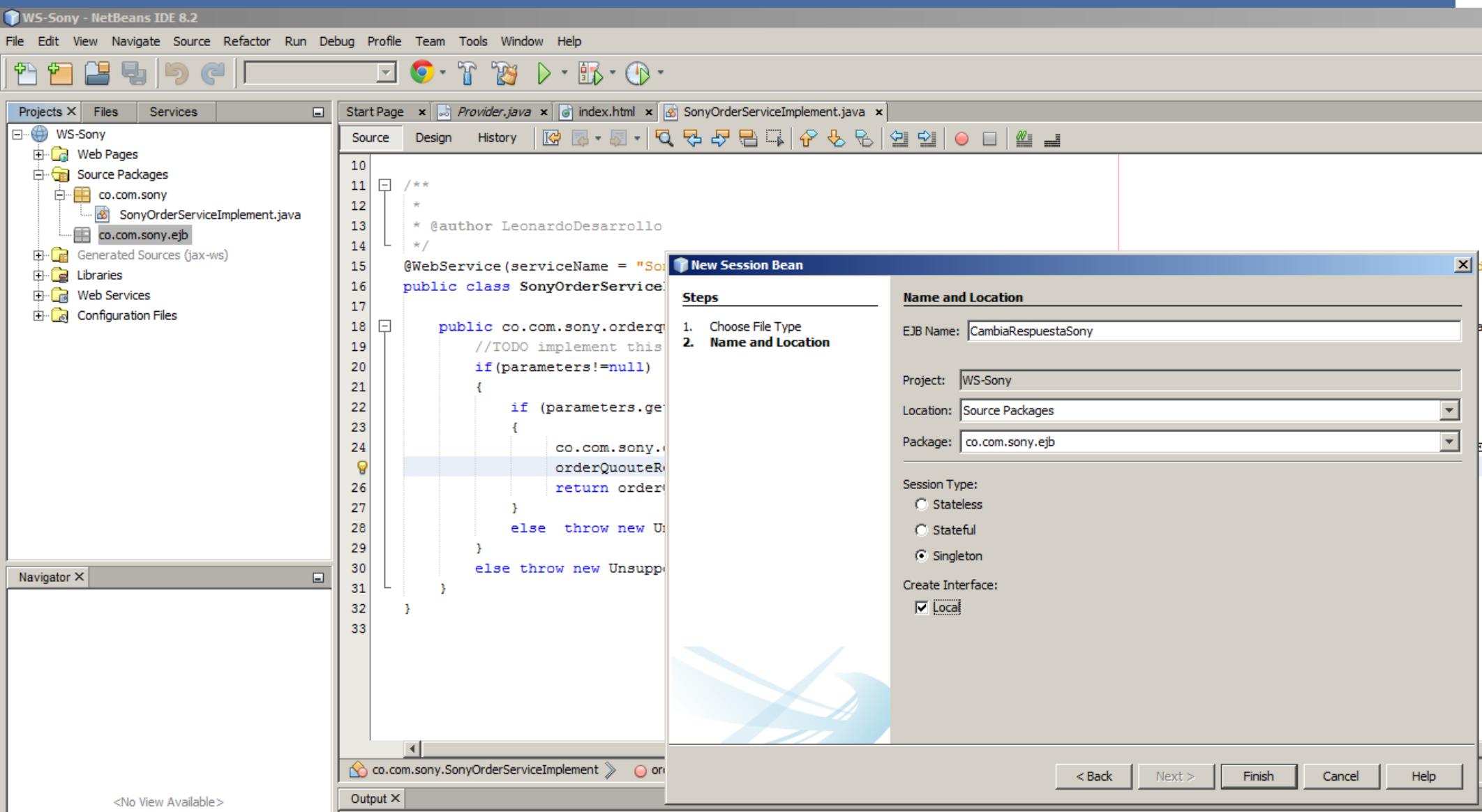
Creamos un nuevo paquete co.com.sony.ejb



Creamos un Session Bean



Nombramos el EJB CambiaRespuestaSony de tipo Singleton y con interface local.



Procedemos a implementar lógica en el EJB.

WS-Sony - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects X Files Services

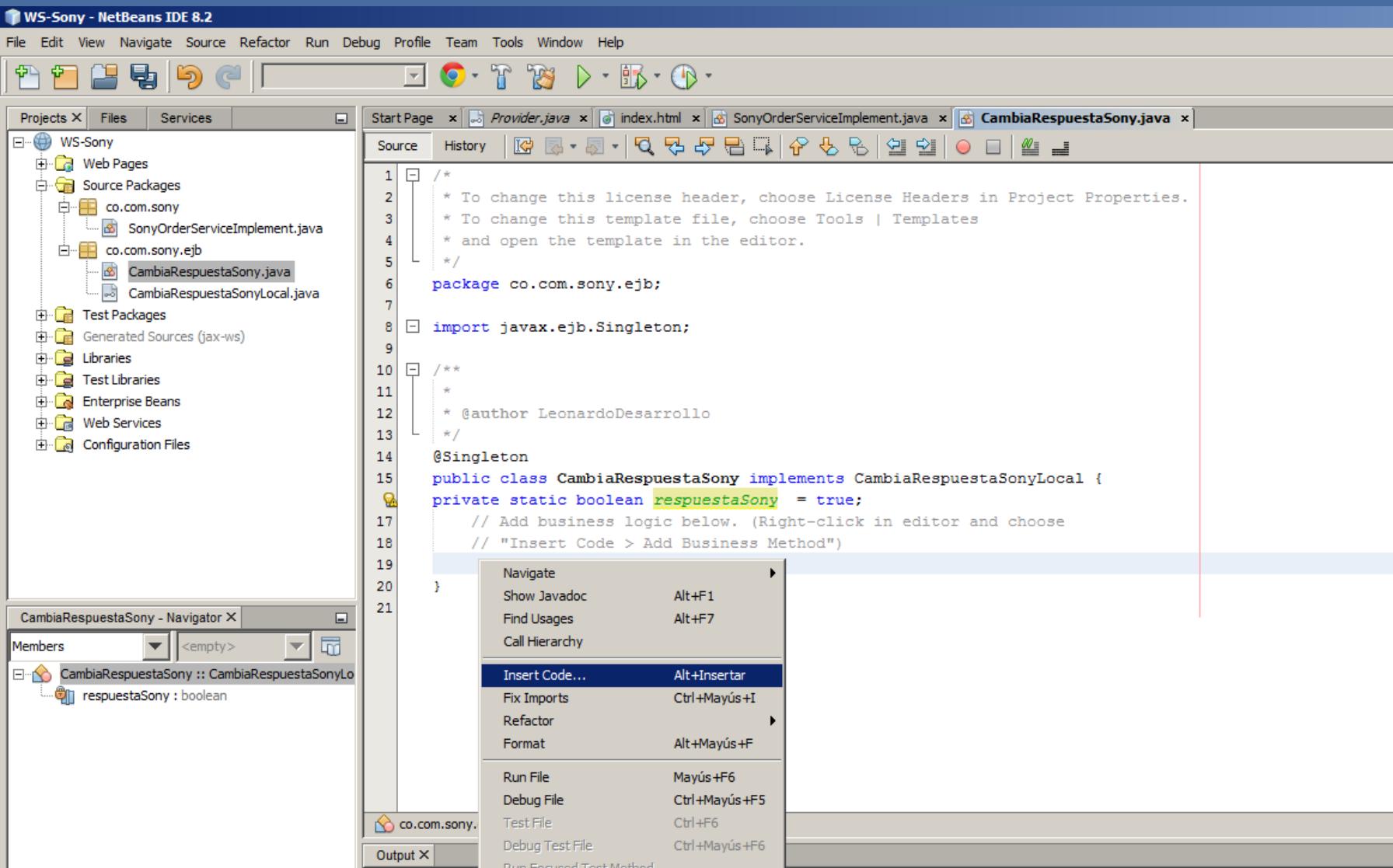
Start Page x Provider.java x index.html x SonyOrderServiceImplement.java x CambiaRespuestaSony.java x

Source History

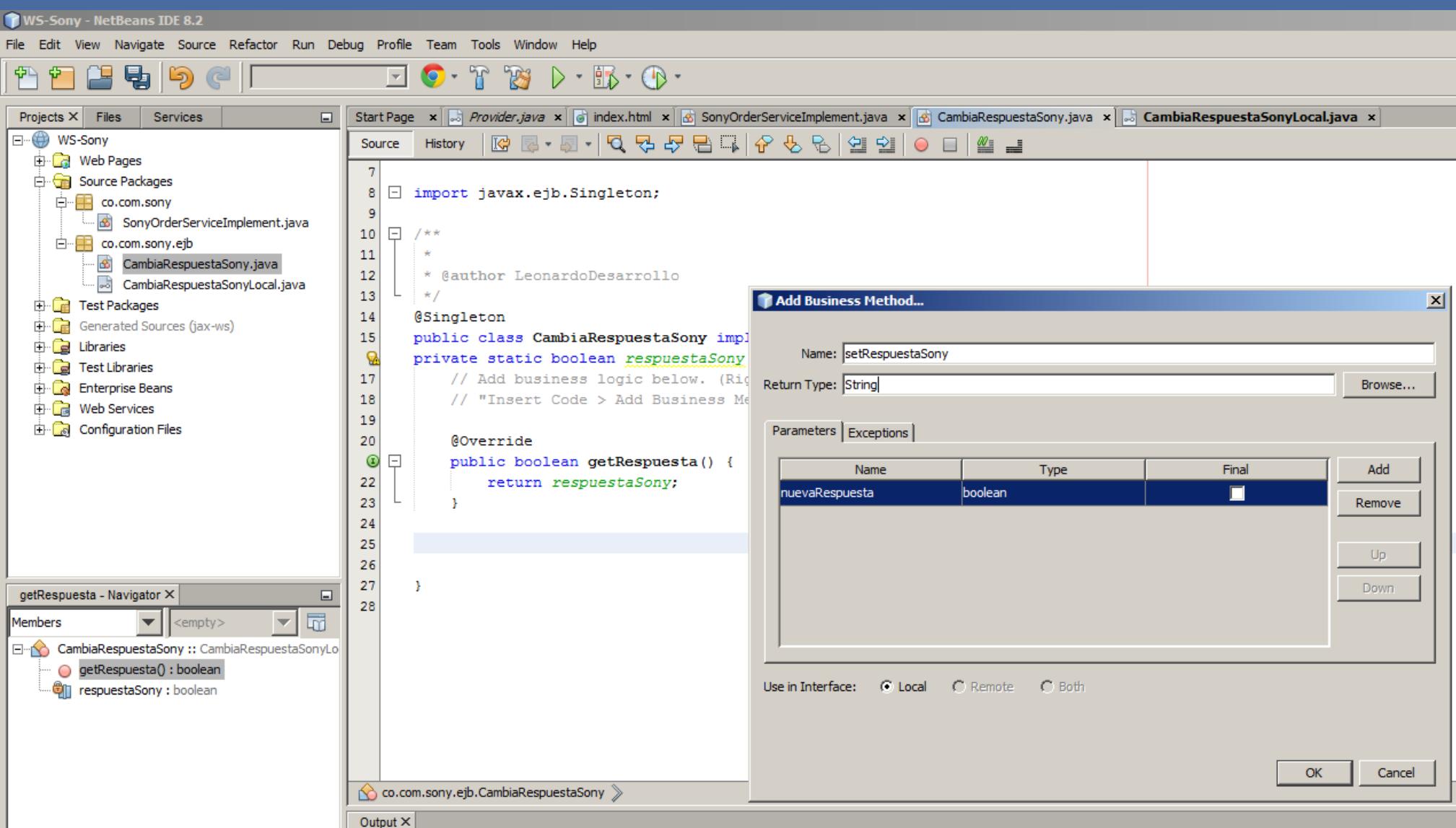
```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6 package co.com.sony.ejb;
7
8 import javax.ejb.Singleton;
9
10 /**
11  *
12  * @author LeonardoDesarrollo
13  */
14 @Singleton
15 public class CambiaRespuestaSony implements CambiaRespuestaSonyLocal {
16
17     // Add business logic below. (Right-click in editor and choose
18     // "Insert Code > Add Business Method")
19
20 }
21
```



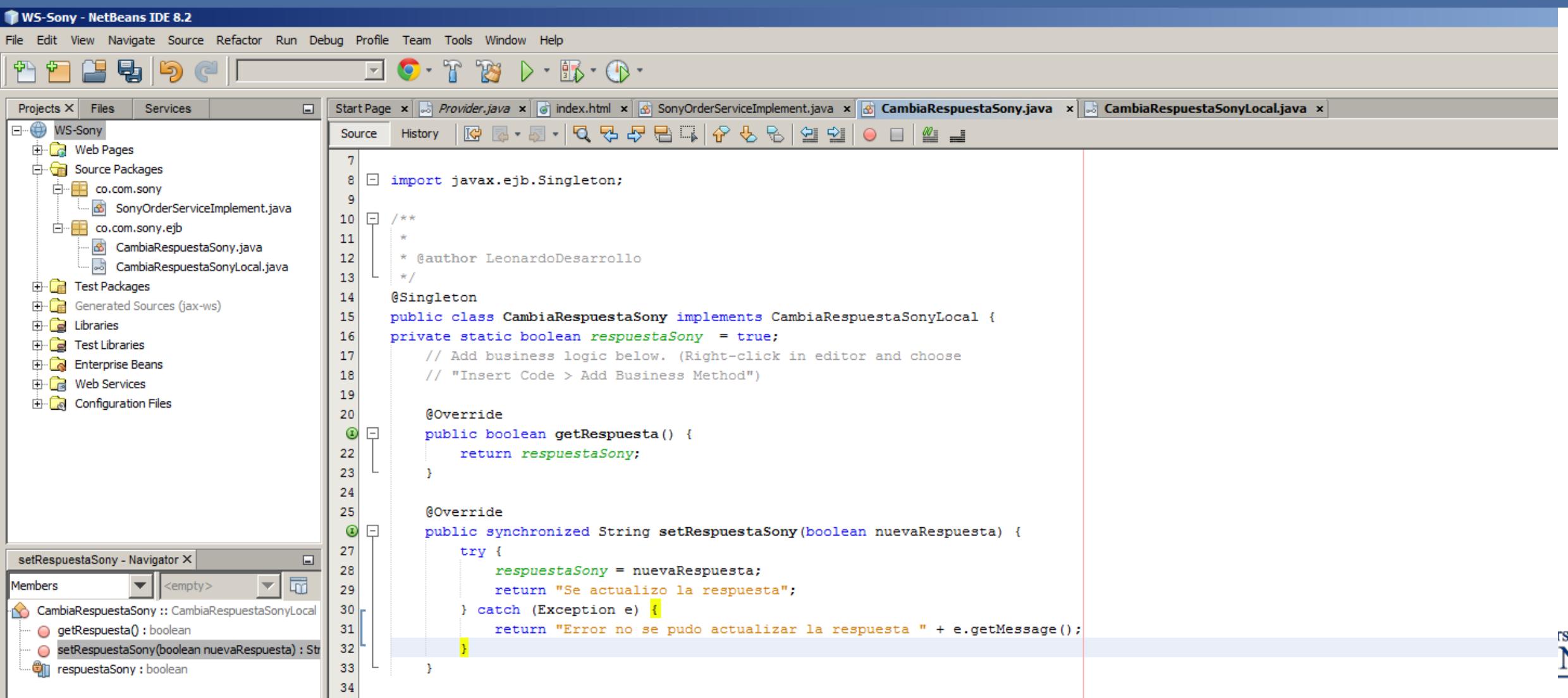
Procedemos a implementar lógica en el EJB. Utilizamos la generación de código para crear métodos de negocio



Procedemos a implementar lógica en el EJB. Utilizamos la generación de código para crear métodos de negocio



Procedemos a implementar lógica en el EJB. Creamos el atributo statico respuestaSony y su implementación.

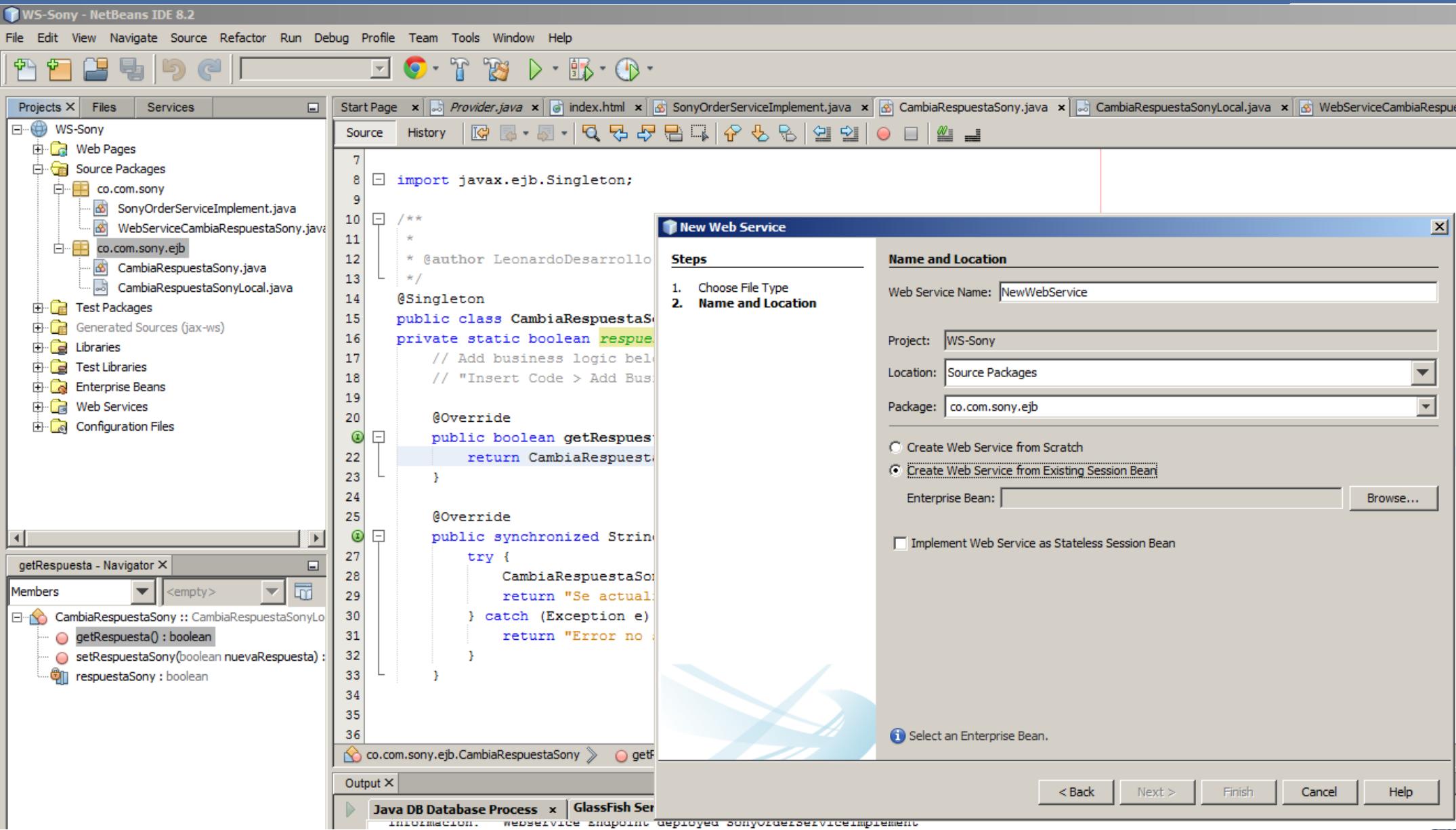


The screenshot shows the NetBeans IDE 8.2 interface with the following details:

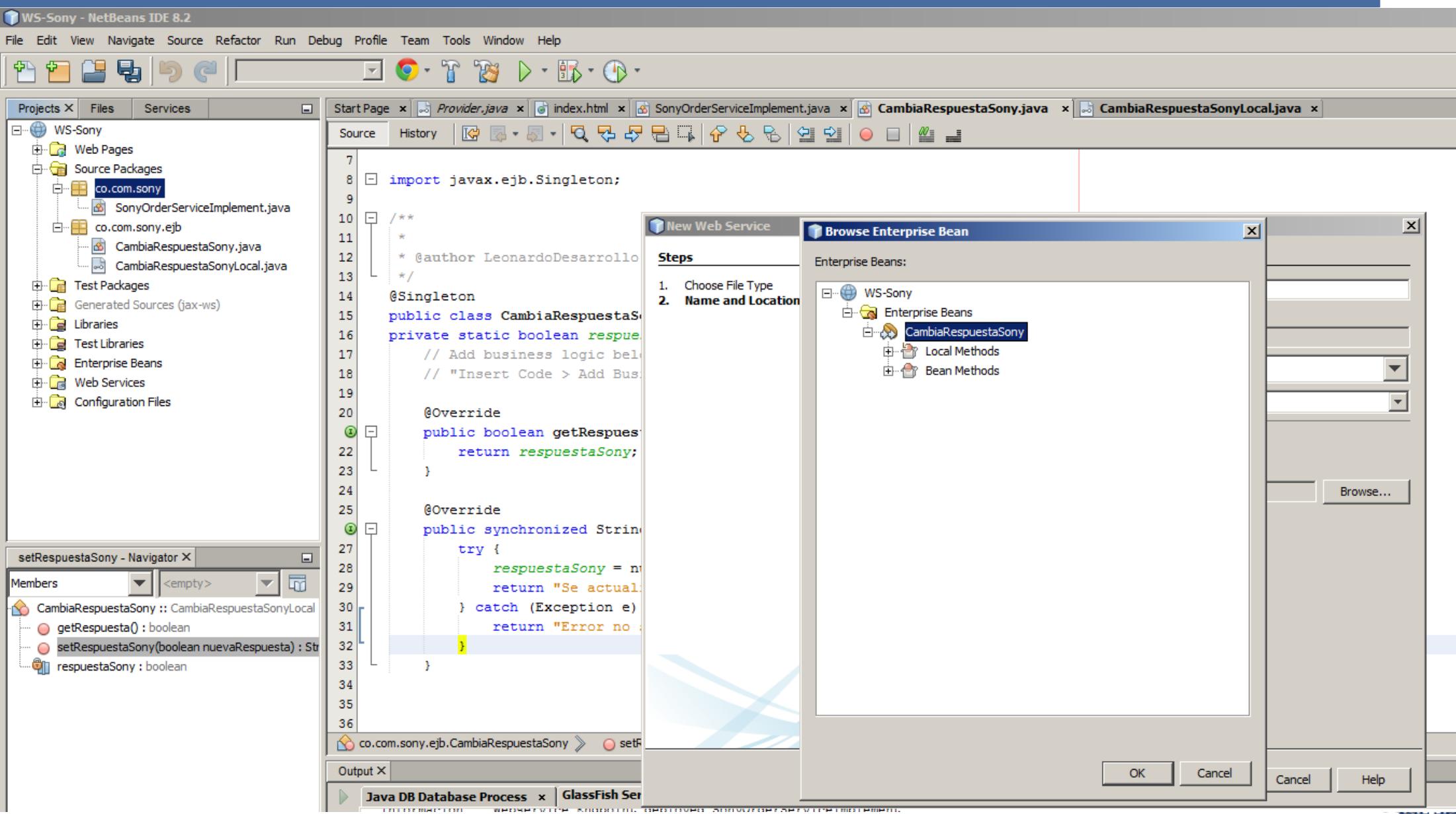
- Title Bar:** WS-Sony - NetBeans IDE 8.2
- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Toolbar:** Standard NetBeans toolbar icons.
- Project Explorer:** Shows the project structure under "WS-Sony".
 - Web Pages
 - Source Packages:
 - co.com.sony: SonyOrderServiceImplement.java
 - co.com.sony.ejb: CambiaRespuestaSony.java, CambiaRespuestaSonyLocal.java
 - Test Packages
 - Generated Sources (jax-ws)
 - Libraries
 - Test Libraries
 - Enterprise Beans
 - Web Services
 - Configuration Files
- Code Editor:** The "CambiaRespuestaSony.java" file is open, showing Java code for an EJB implementation.

```
7 import javax.ejb.Singleton;
8
9 /**
10  * 
11  * @author LeonardoDesarrollo
12  */
13 @Singleton
14 public class CambiaRespuestaSony implements CambiaRespuestaSonyLocal {
15     private static boolean respuestaSony = true;
16
17     // Add business logic below. (Right-click in editor and choose
18     // "Insert Code > Add Business Method")
19
20     @Override
21     public boolean getRespuesta() {
22         return respuestaSony;
23     }
24
25     @Override
26     public synchronized String setRespuestaSony(boolean nuevaRespuesta) {
27         try {
28             respuestaSony = nuevaRespuesta;
29             return "Se actualizo la respuesta";
30         } catch (Exception e) {
31             return "Error no se pudo actualizar la respuesta " + e.getMessage();
32         }
33     }
34 }
```
- Navigator:** The "setRespuestaSony - Navigator X" window shows the members of the CambiaRespuestaSony class:
 - Members dropdown: CambiaRespuestaSony :: CambiaRespuestaSonyLocal
 - Members list:
 - getRespuesta(): boolean
 - setRespuestaSony(boolean nuevaRespuesta): String
 - respuestaSony: boolean
- Status Bar:** rsidad NA

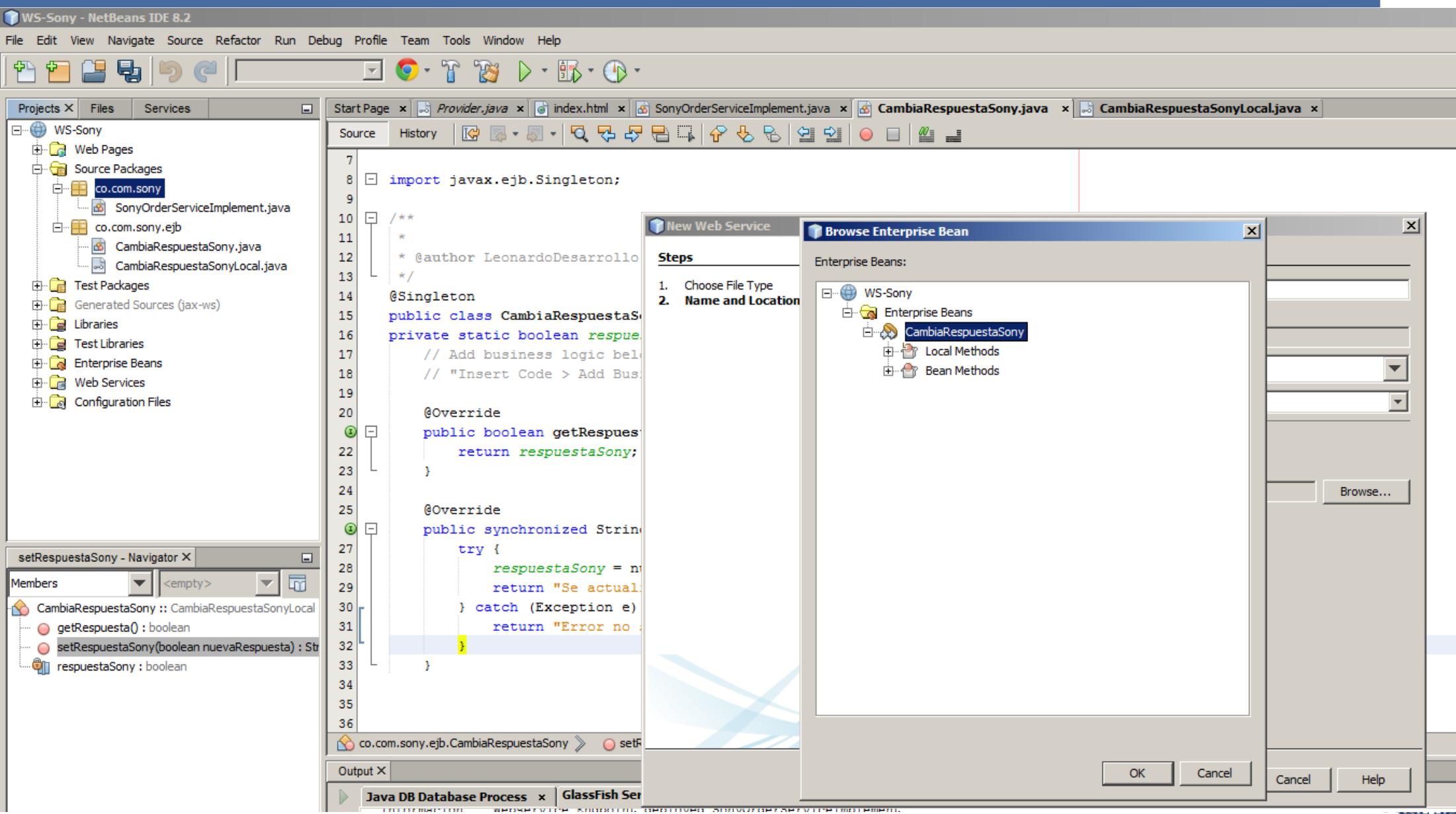
Creamos un nuevo WS basado en un Session Bean existente



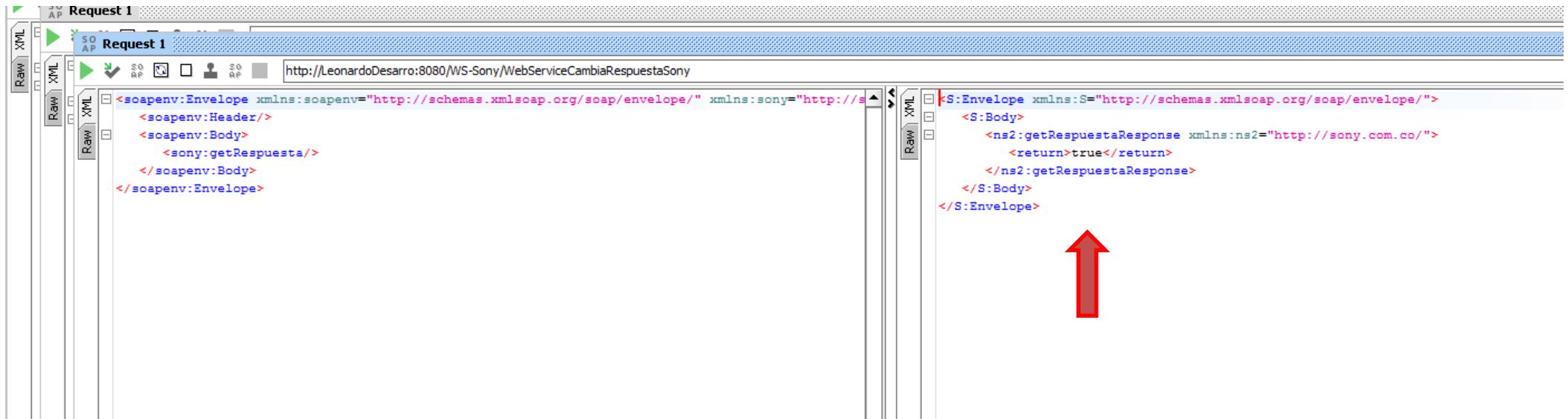
Seleccionamos el EJB para el WS



Seleccionamos el EJB para el WS



Probamos la operación getRespuesta



The screenshot shows a SOAP message exchange interface. On the left, a navigation tree lists service components: SonyOrderService, WebServiceCambiaRespuestaSony, and WebServiceCambiaRespuestaSonyPortBinding. Under WebServiceCambiaRespuestaSonyPortBinding, there are operations: getRespuesta, setRespuestaSony, and another getRespuesta entry. The main area displays two messages: "Request 1" and "Response 1". The "Request 1" pane shows a SOAP envelope with a header and a body containing a "sony:getRespuesta" call. The "Response 1" pane shows a SOAP envelope with a body containing a "ns2:getRespuestaResponse" element, which has a "return" attribute set to "true". A red arrow points upwards from the "return" element towards the top of the response body.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sony="http://sony.com.co">
    <soapenv:Header/>
    <soapenv:Body>
        <sony:getRespuesta/>
    </soapenv:Body>
</soapenv:Envelope>
```



```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:getRespuestaResponse xmlns:ns2="http://sony.com.co/">
            <return>true</return>
        </ns2:getRespuestaResponse>
    </S:Body>
</S:Envelope>
```

Probamos la operación setRespuesta

SoapUI 5.3.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Search Forum

SonyOrderService

- SonyOrderServiceBinding
 - orderQuoute
 - Request 1
- WebServiceCambiaRespuestaSony
- WebServiceCambiaRespuestaSonyPortBinding
 - getRespuesta
 - Request 1
 - setRespuestaSony
 - Request 1

Request 1

Request 1

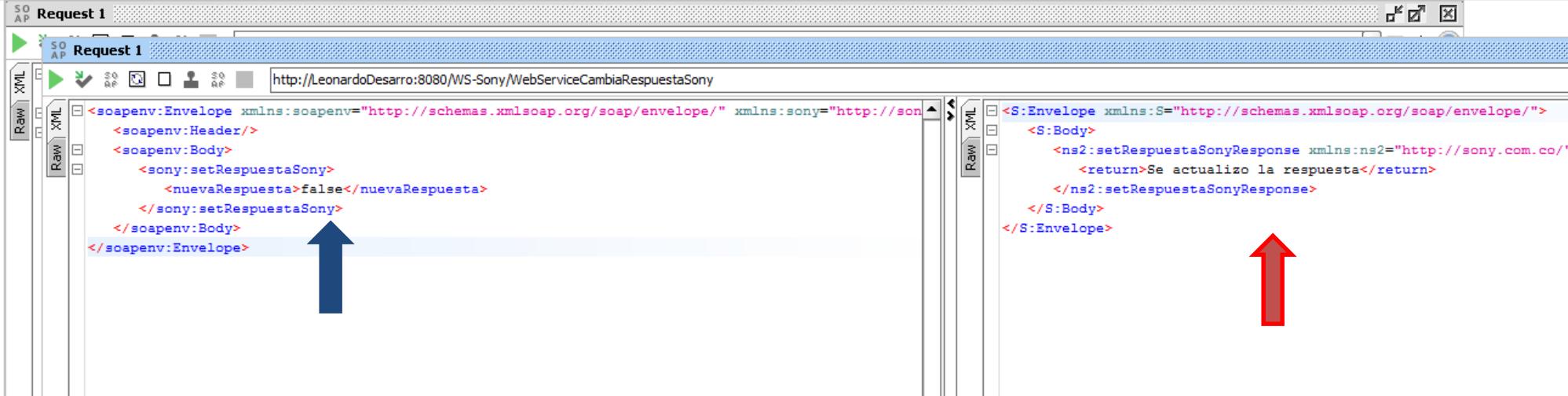
http://LeonardoDesarro:8080/WS-Sony/WebServiceCambiaRespuestaSony

Raw XML Raw XML Raw XML Raw XML

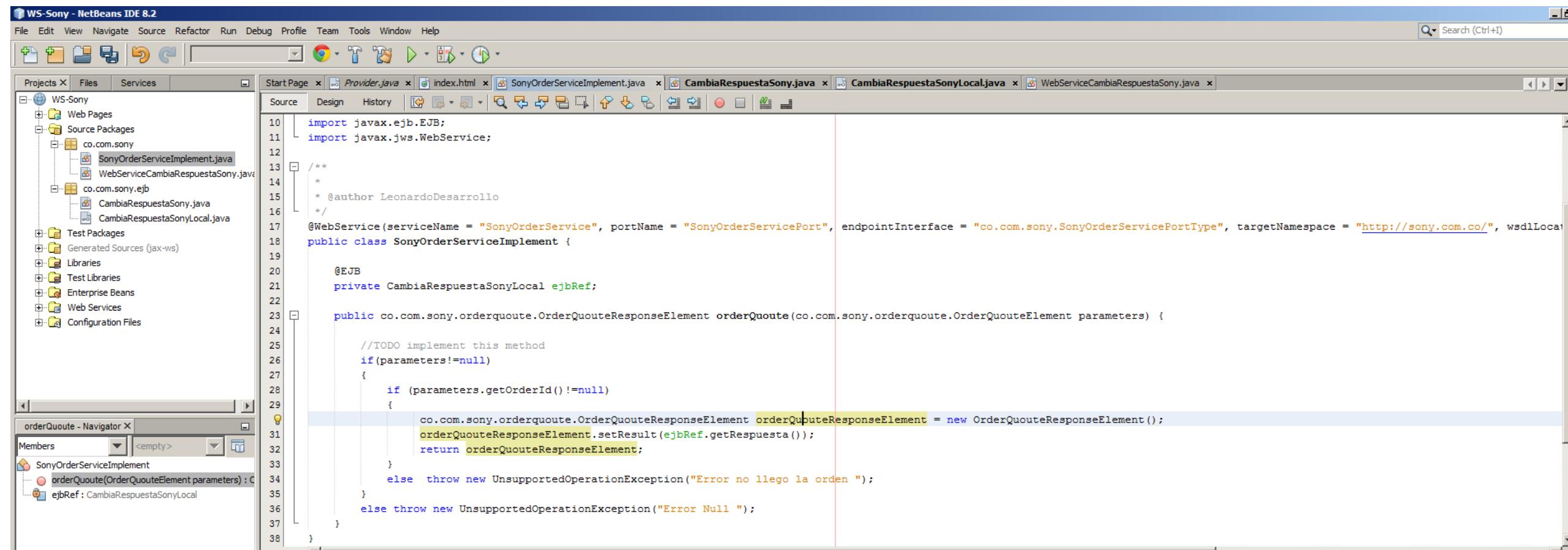
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sony="http://sony.com.co/">  
    <soapenv:Header/>  
    <soapenv:Body>  
        <sony:setRespuestaSony>  
            <nuevaRespuesta>false</nuevaRespuesta>  
        </sony:setRespuestaSony>  
    </soapenv:Body>  
</soapenv:Envelope>
```

Raw XML Raw XML Raw XML Raw XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">  
    <S:Body>  
        <ns2:setRespuestaSonyResponse xmlns:ns2="http://sony.com.co/">  
            <return>Se actualizo la respuesta</return>  
        </ns2:setRespuestaSonyResponse>  
    </S:Body>  
</S:Envelope>
```



Modificamos la implementación del servicio Ordenes de Sony Para obtenga la respuesta del EJB y no el valor quemado.



The screenshot shows the NetBeans IDE 8.2 interface with the following details:

- Title Bar:** WS-Sony - NetBeans IDE 8.2
- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Toolbar:** Standard NetBeans toolbar icons.
- Project Explorer:** Shows the project structure under "WS-Sony".
- Code Editor:** Displays the Java code for `SonyOrderServiceImplement.java`. The code implements a Web Service with a single method `orderQuoute` that returns a response element. A red vertical line highlights the line where the response is set.

```
10 import javax.ejb.EJB;
11 import javax.jws.WebService;
12
13 /**
14 * @author LeonardoDesarrollo
15 */
16
17 @WebService(serviceName = "SonyOrderService", portName = "SonyOrderServicePort", endpointInterface = "co.com.sony.SonyOrderServicePortType", targetNamespace = "http://sony.com.co/", wsdlLoca
18 public class SonyOrderServiceImplement {
19
20     @EJB
21     private CambiaRespuestaSonyLocal ejbRef;
22
23     public co.com.sony.orderquoute.OrderQuouteResponseElement orderQuoute(co.com.sony.orderquoute.OrderQuouteElement parameters) {
24
25         //TODO implement this method
26         if(parameters!=null)
27         {
28             if (parameters.getOrderId() !=null)
29             {
30                 co.com.sony.orderquoute.OrderQuouteResponseElement orderQuouteResponseElement = new OrderQuouteResponseElement();
31                 orderQuouteResponseElement.setResult(ejbRef.getRespuesta());
32                 return orderQuouteResponseElement;
33             }
34             else throw new UnsupportedOperationException("Error no llego la orden ");
35         }
36         else throw new UnsupportedOperationException("Error Null ");
37     }
38 }
```

- Navigator:** Shows the members of the `SonyOrderServiceImplement` class, including the `orderQuoute` method and the `ejbRef` field.

Probamos la operación OrderSony

SoapUI 5.3.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy

SonyOrderService

- SonyOrderServiceBinding
 - orderQuoute
 - SO Request 1
- WebServiceCambiaRespuestaSony
- WebServiceCambiaRespuestaSonyPortBinding
 - getRespuesta
 - setRespuestaSony
 - SO Request 1

Request 1

Submit request to specified endpoint URL (Alt-Enter) http://LeonardoDesarro:8080/WS-Sony/SonyOrderService

Raw XML

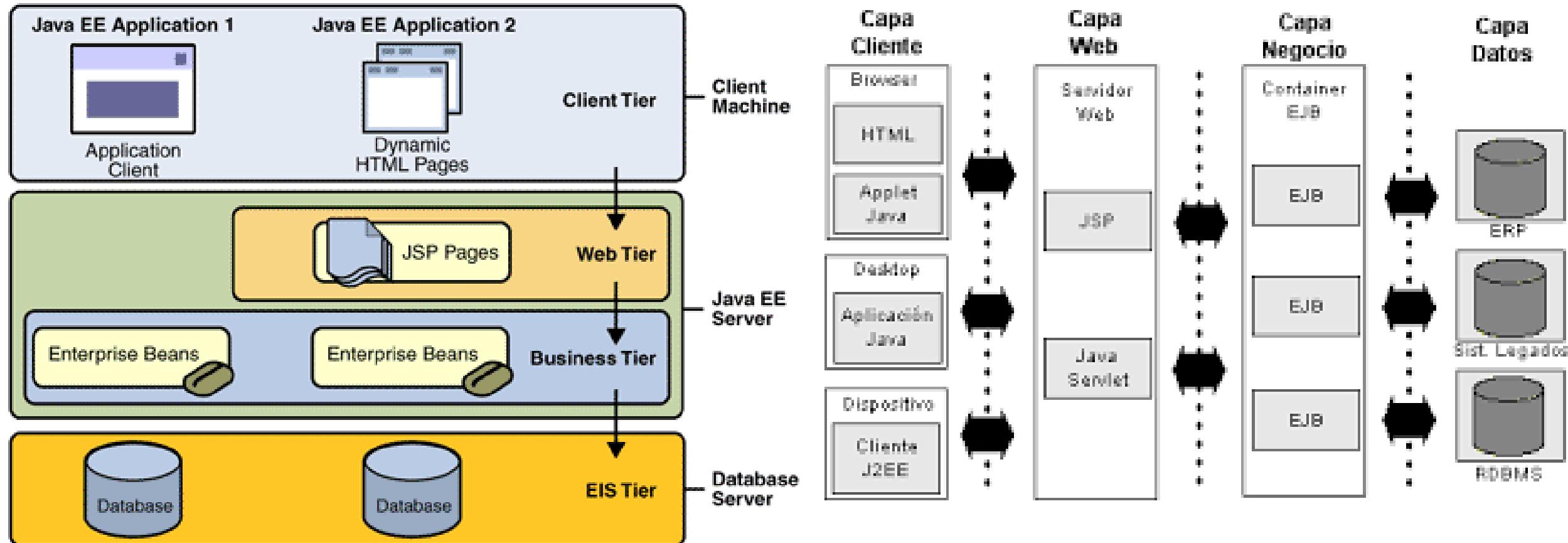
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
  <soapenv:Body>
    <ord:orderQuouteElement>
      <orderId>123</orderId>
    </ord:orderQuouteElement>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:orderQuouteResponseElement xmlns:ns2="http://sony.com.co/orderQuoute">
      <result>false</result>
    </ns2:orderQuouteResponseElement>
  </S:Body>
</S:Envelope>
```



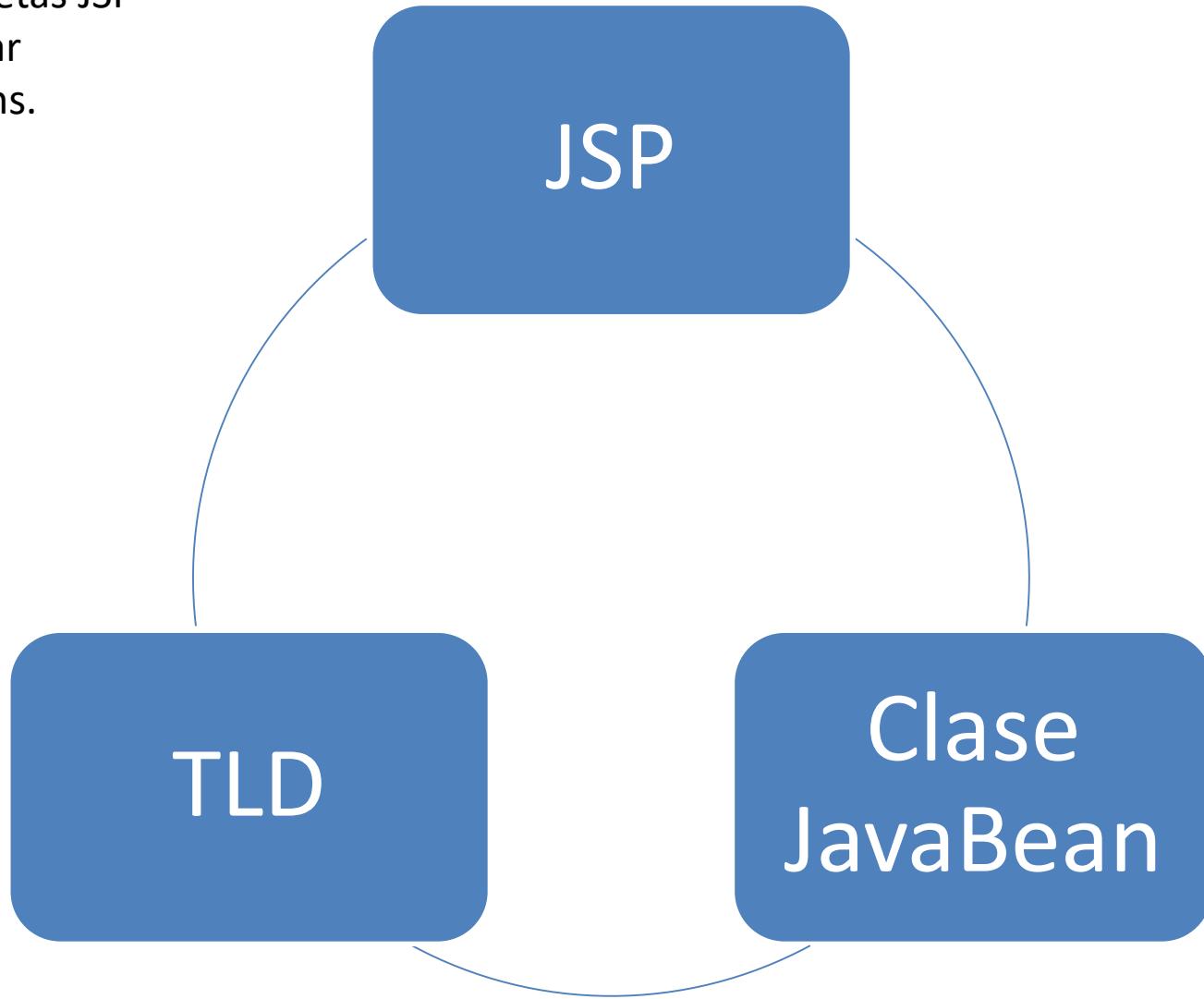
Arquitecturas Web

Arquitectura WEB



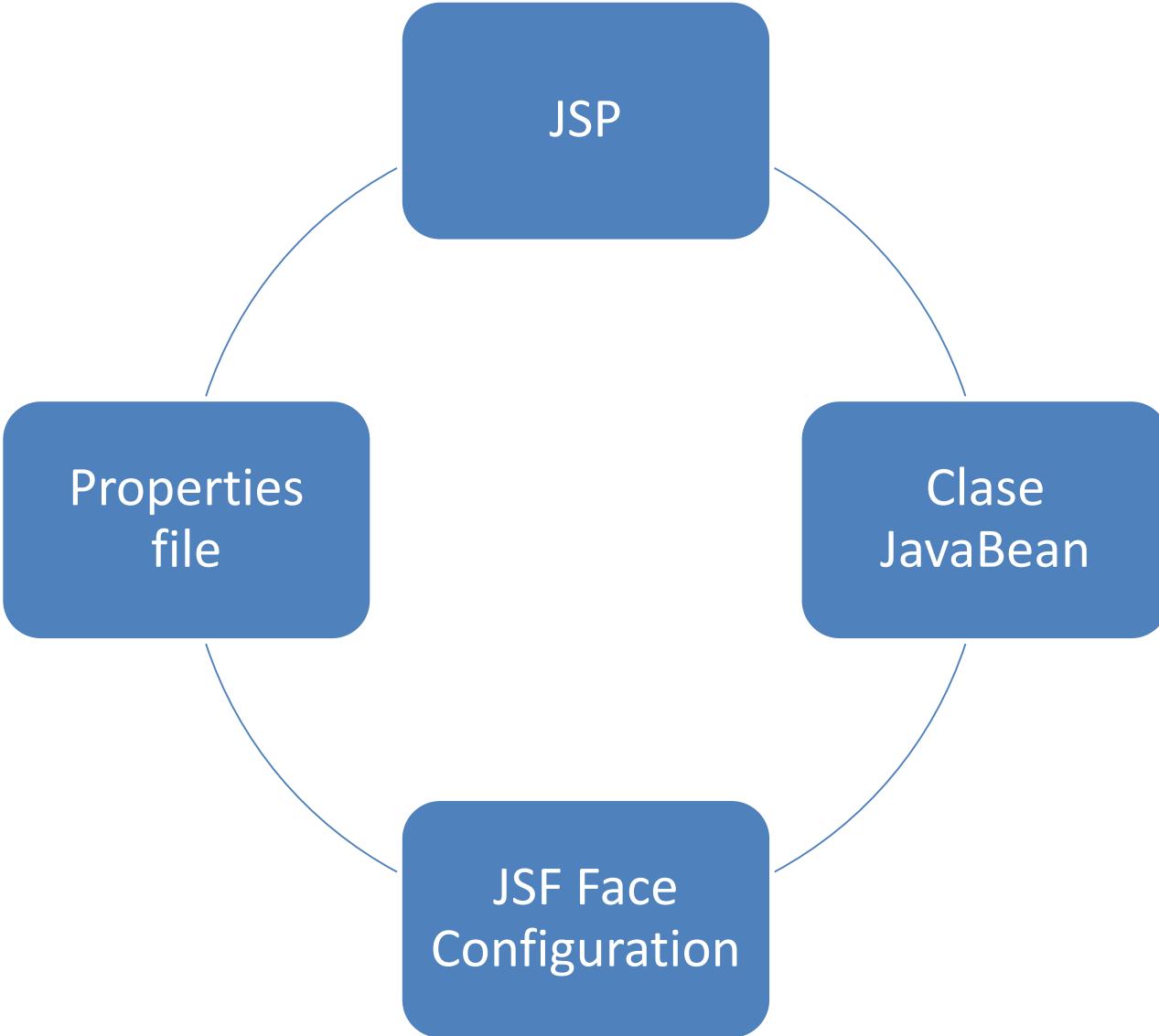
Actores con JSP

Tag Library Descriptor: etiquetas JSP útiles que permite encapsular funcionalidades de los Beans.



Un **JavaBean** (o también conocido simplemente como Bean) es una clase simple en Java que cumple con ciertas normas con los nombres de sus propiedades y métodos.

Actores con JSF

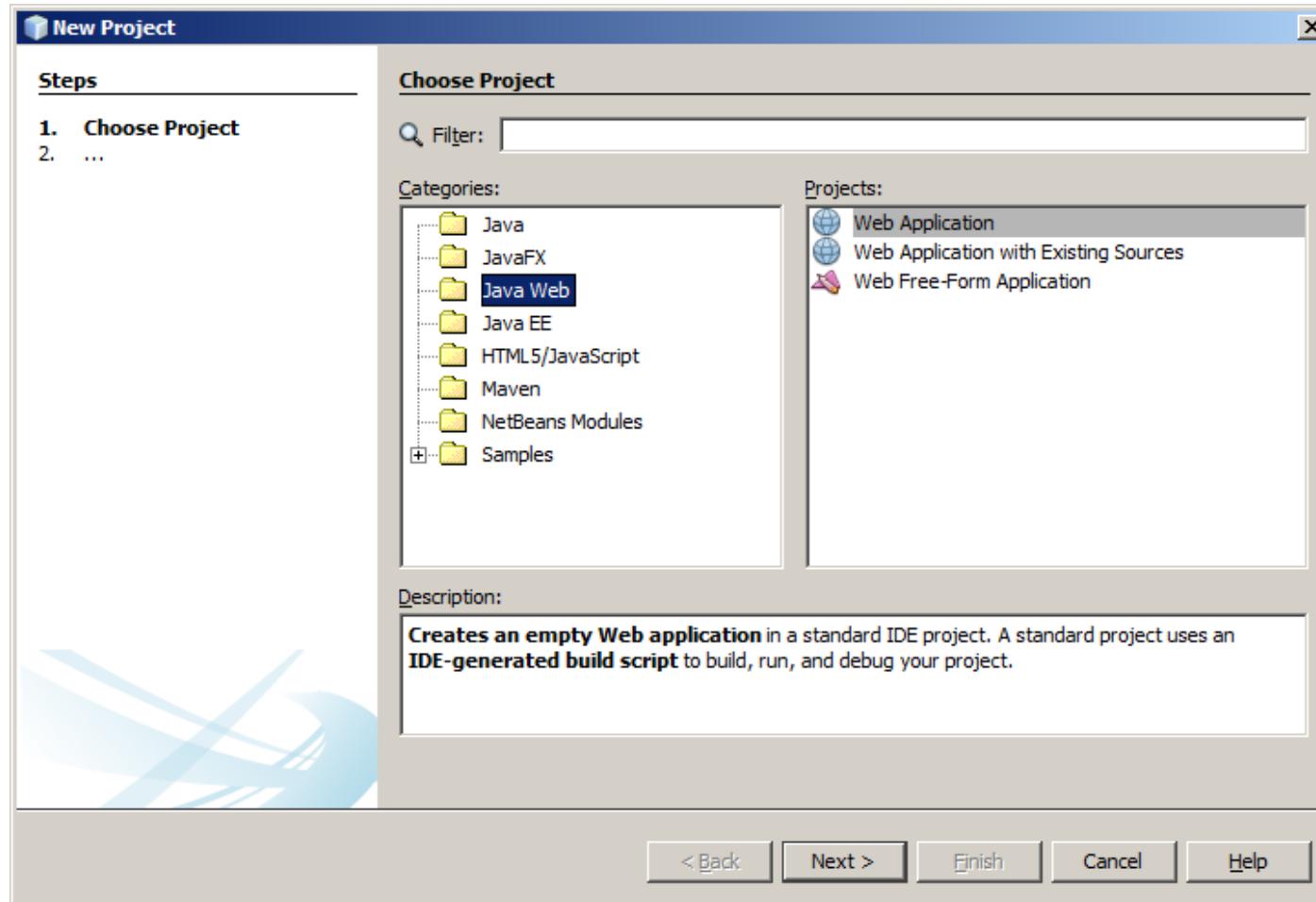


Un **JavaBean** (o también conocido simplemente como Bean) es una clase simple en Java que cumple con ciertas normas con los nombres de sus propiedades y métodos.

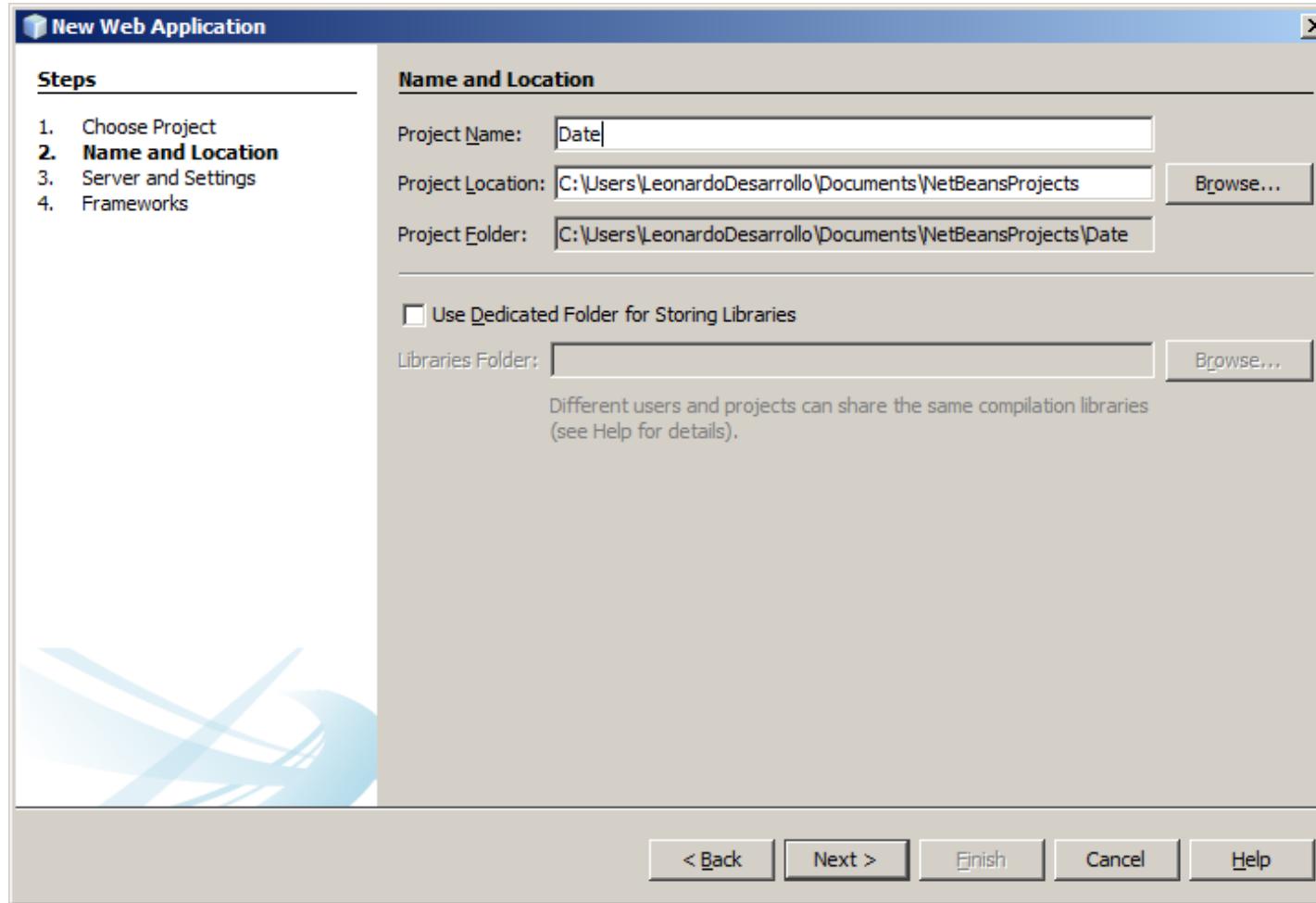
JSP proyecto Date.

Consiste en la construcción de una página JSP que en una relación de dependencia de un combo desplegable que contiene diferentes idiomas, nos pueda indicar la hora actual en el idioma acorde a la selección.

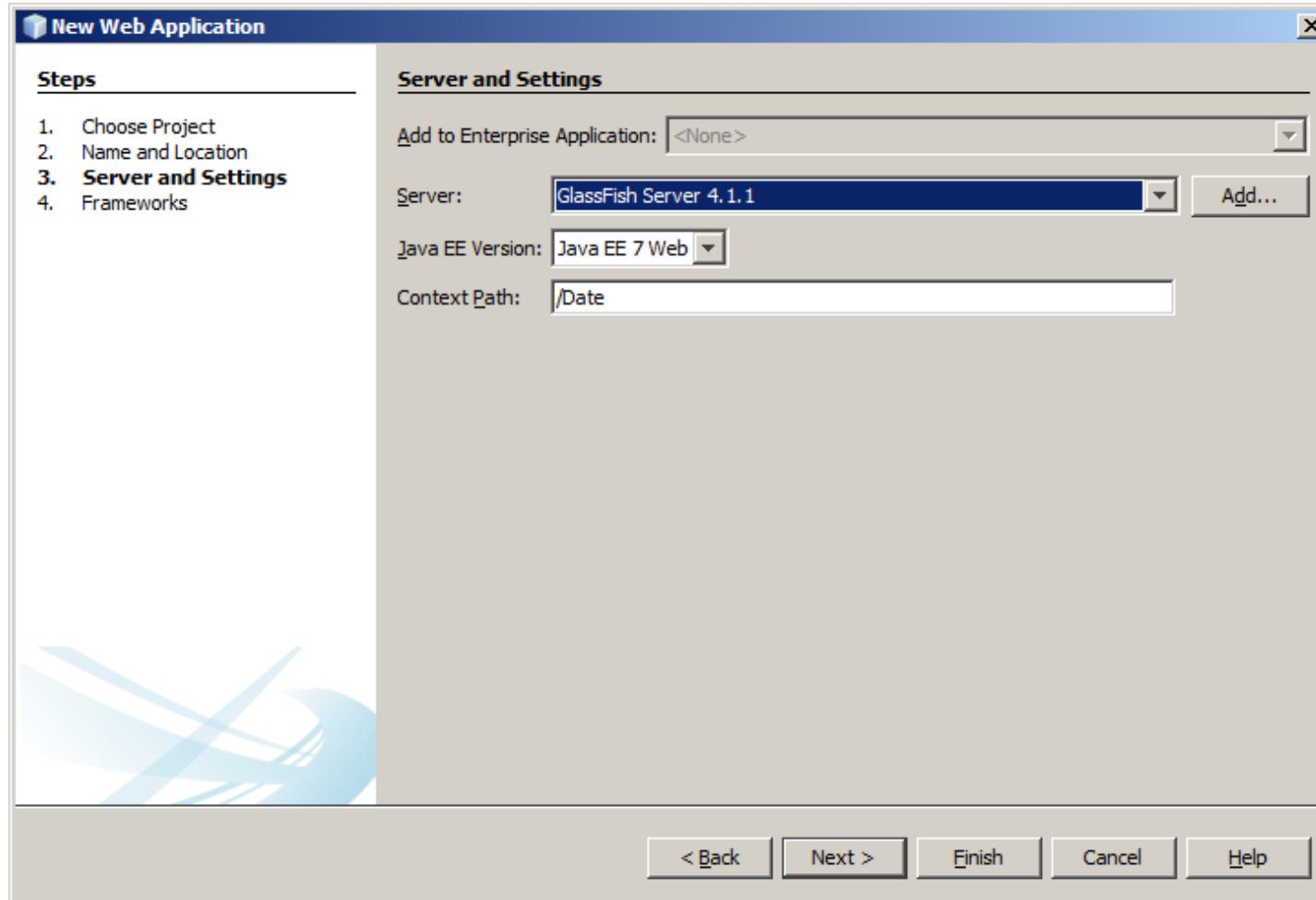
Creamos nuevo proyecto Web



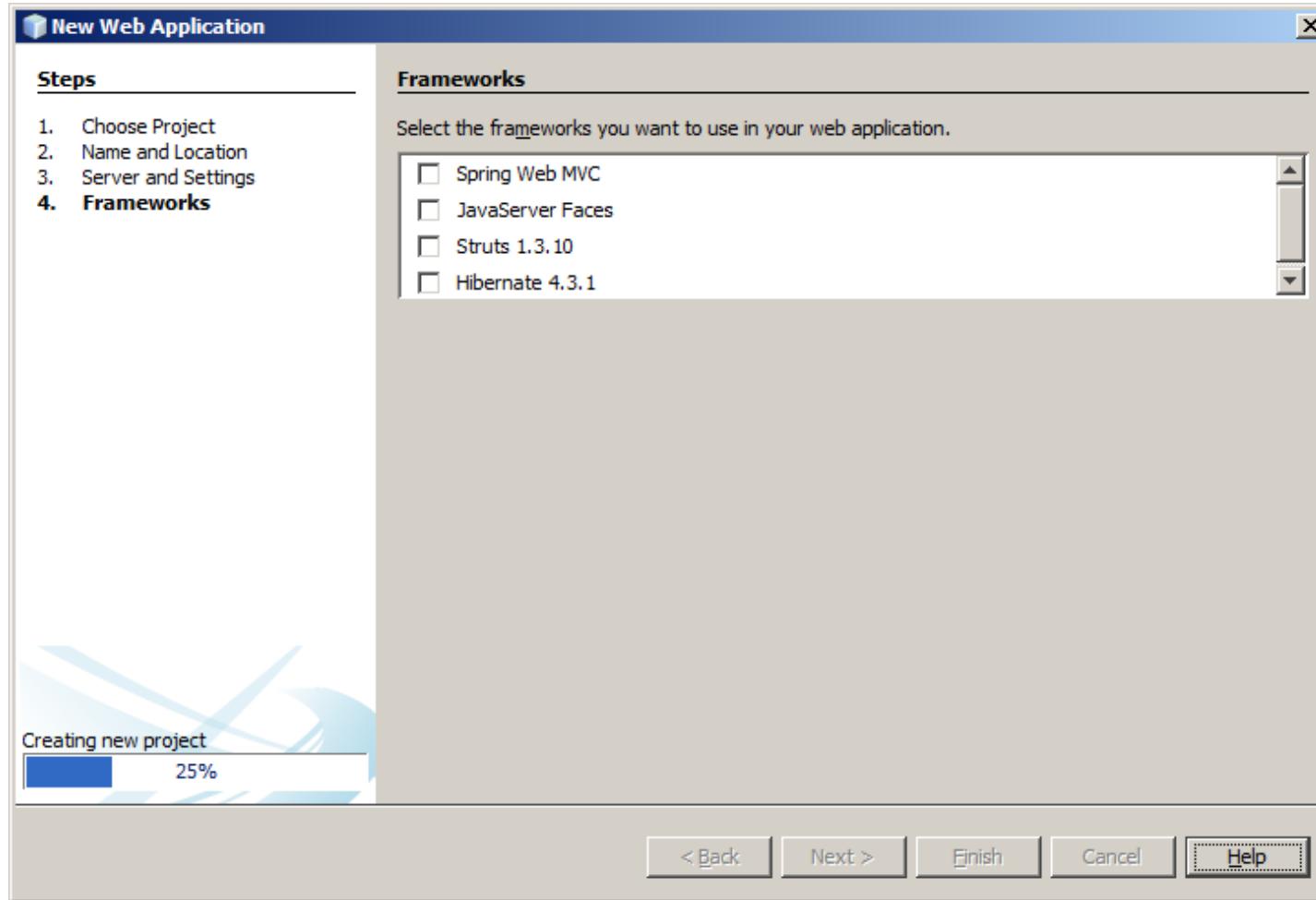
Asignamos nombre del proyecto



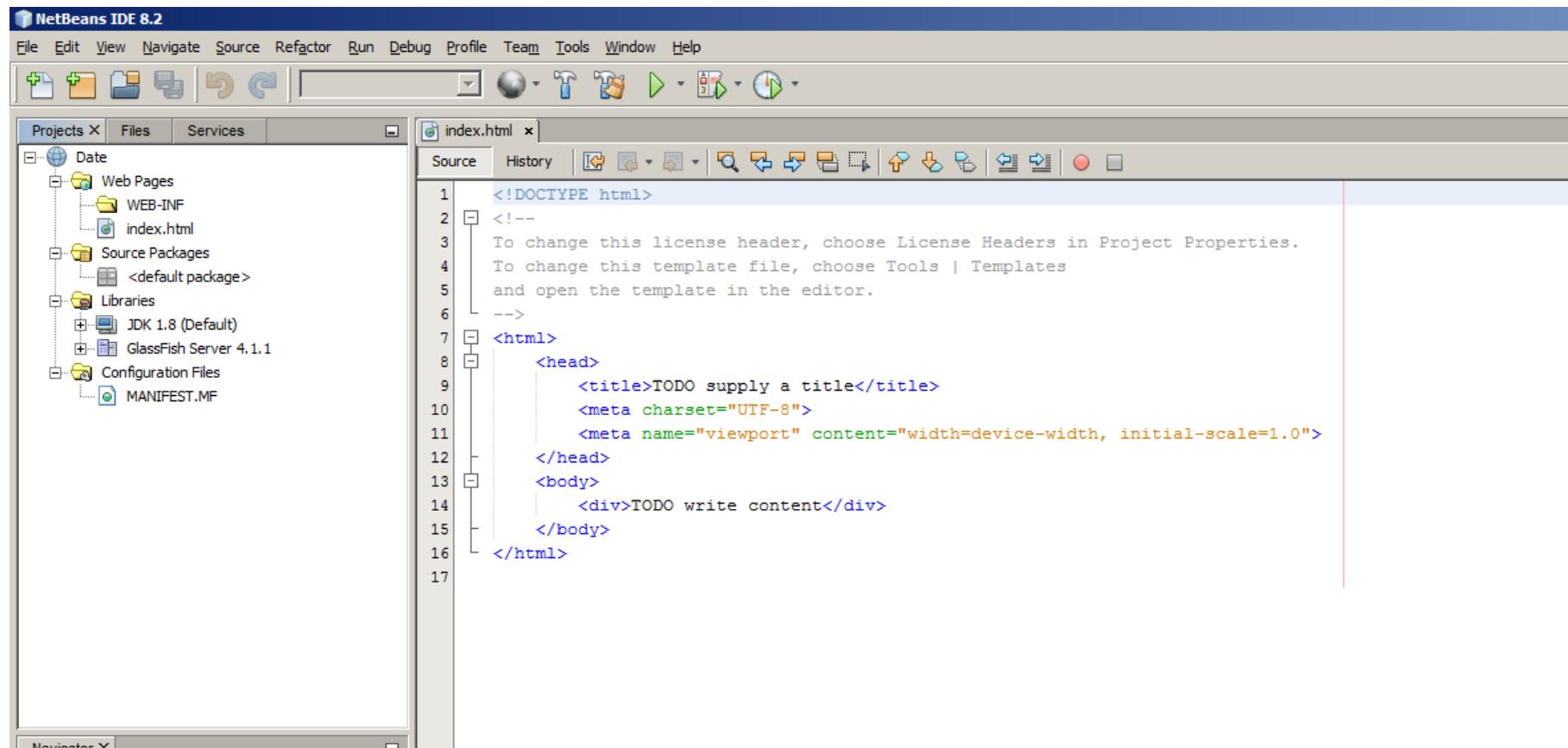
Configuramos el servidor



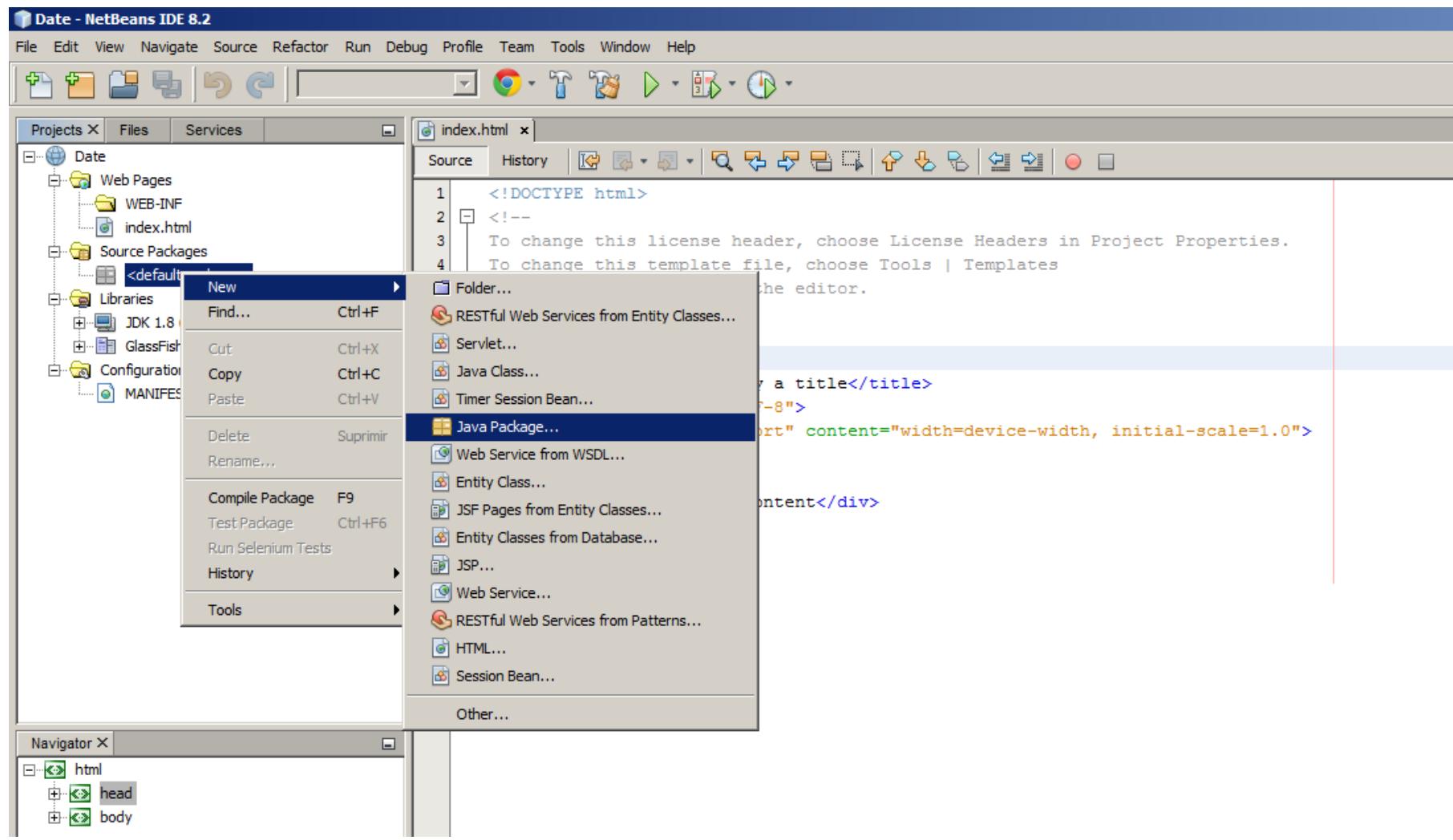
No Seleccionamos ningún Framework



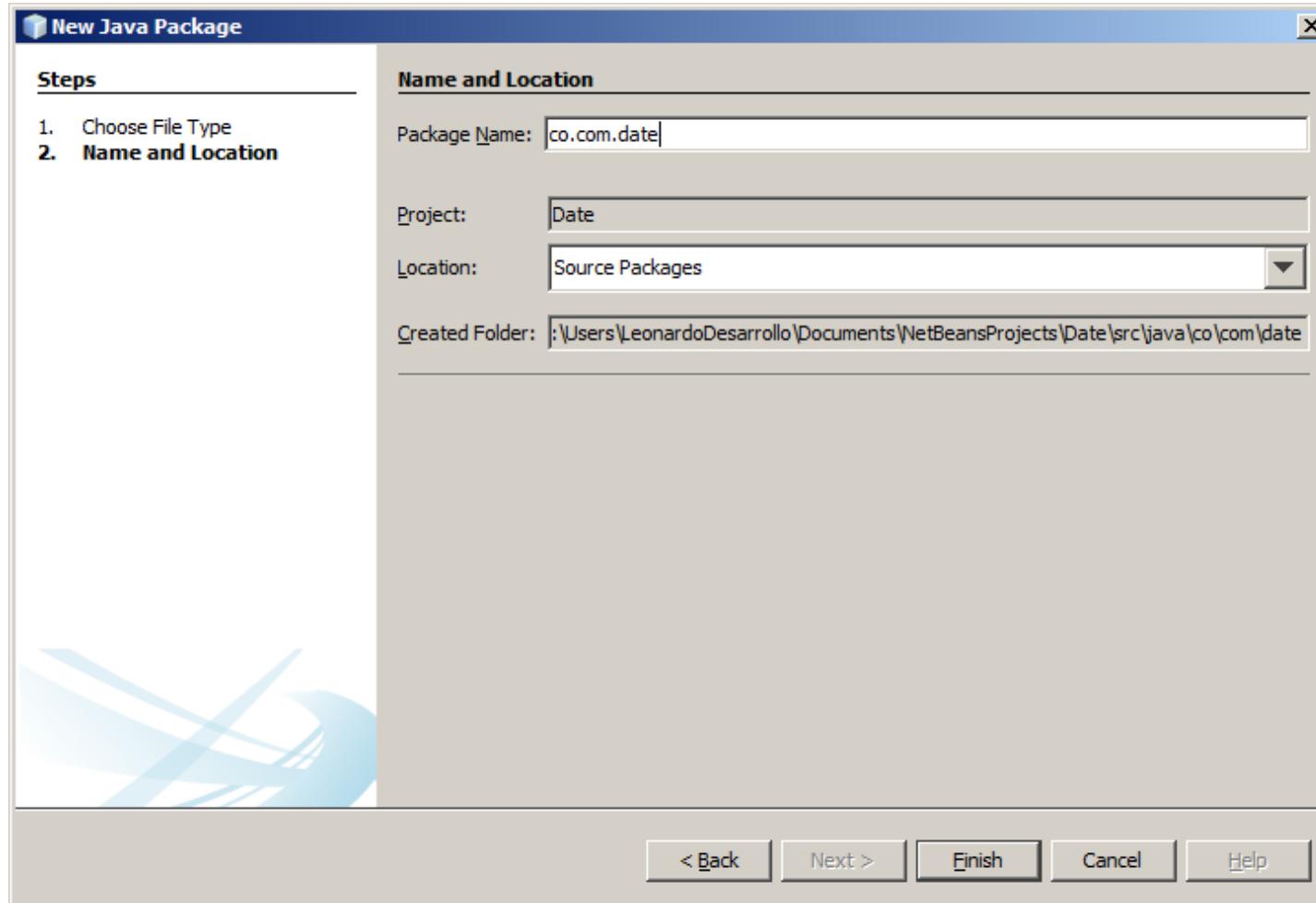
Se crea el proyecto Date



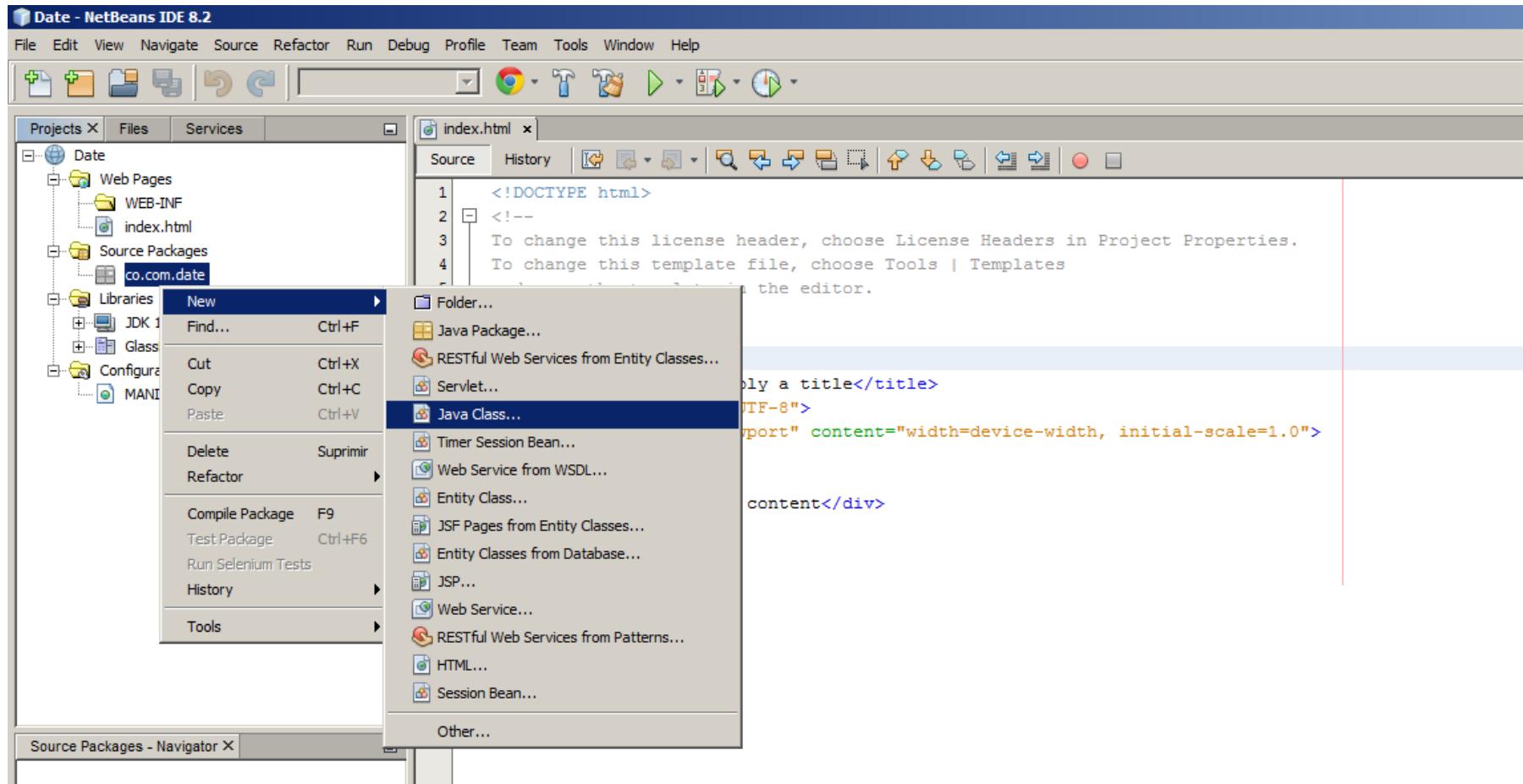
Creamos el paquete



Creamos el paquete co.com.date



Creamos una nueva clase MyDate

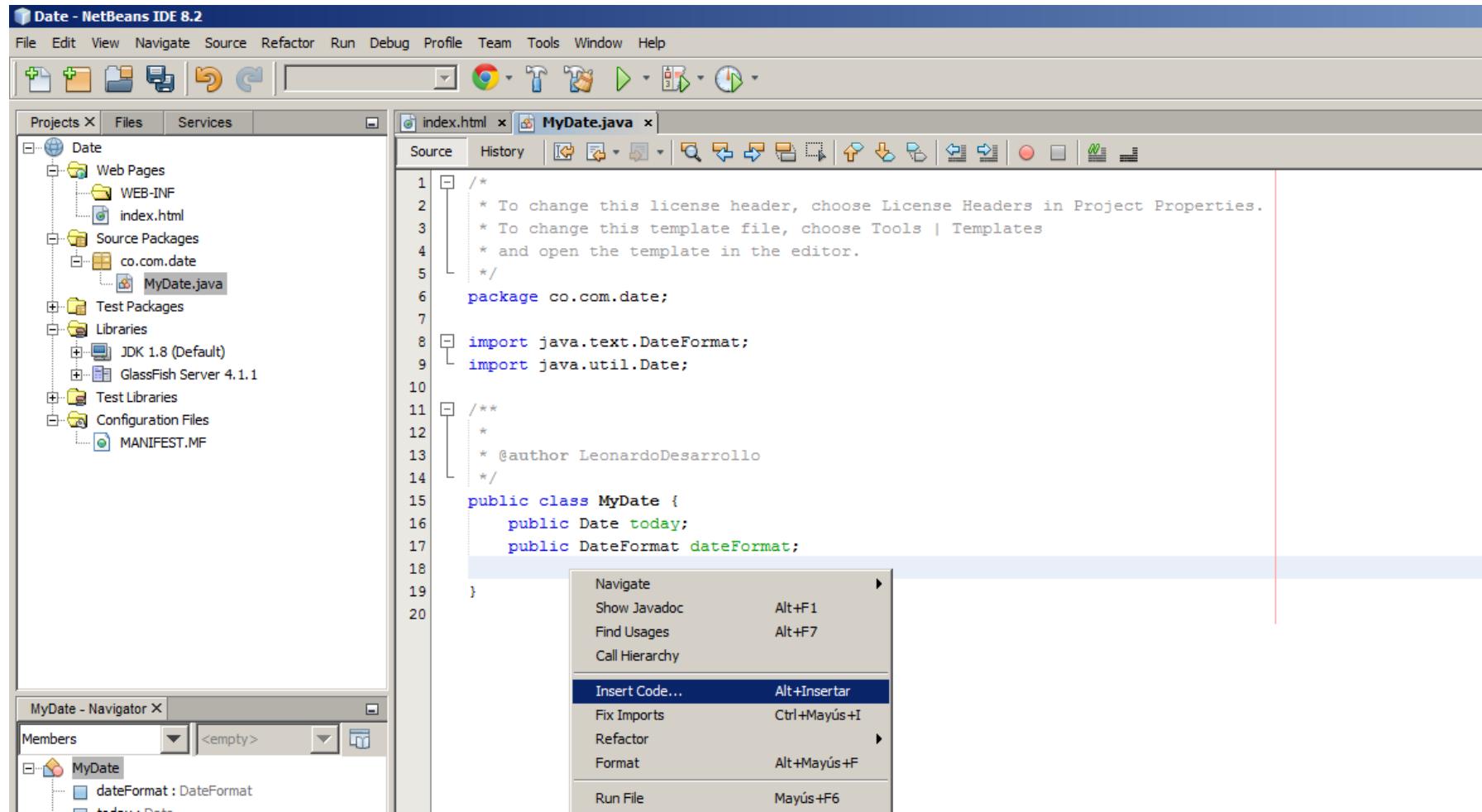


Creamos una nueva clase

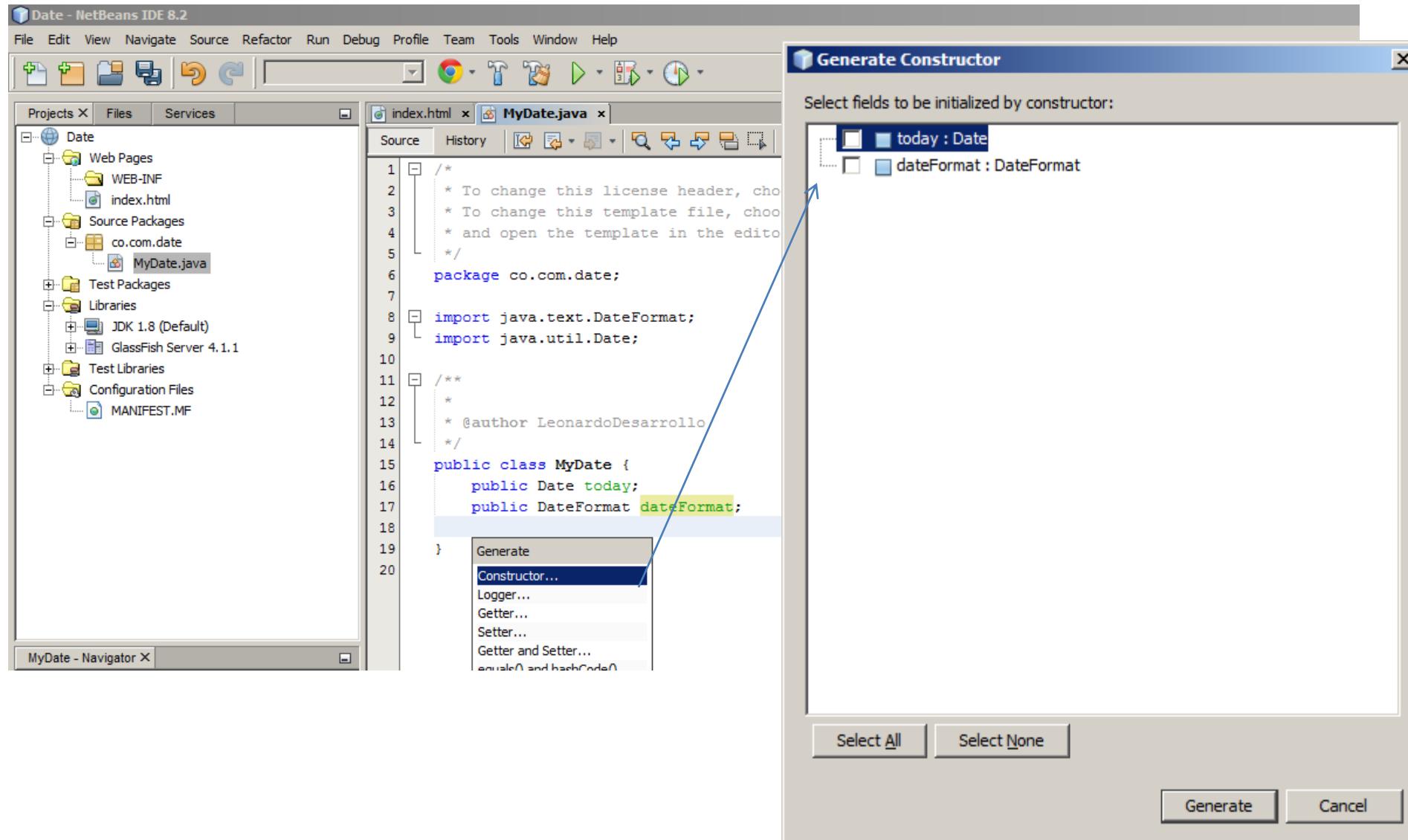
The screenshot shows the NetBeans IDE 8.2 interface. The title bar reads "Date - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar has various icons for file operations like New, Open, Save, and Run. The Projects tab in the left sidebar shows a project named "Date" containing "Web Pages", "Source Packages" (with a file "MyDate.java" selected), "Test Packages", "Libraries" (including "JDK 1.8 (Default)" and "GlassFish Server 4.1.1"), "Test Libraries", and "Configuration Files". The Navigator tab shows an empty members list. The main editor window displays the code for "MyDate.java":

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package co.com.date;
7
8  import java.text.DateFormat;
9  import java.util.Date;
10
11 /**
12 *
13 * @author LeonardoDesarrollo
14 */
15 public class MyDate {
16     public Date today;
17     public DateFormat dateFormat;
18 }
19
20 }
```

Utilizamos la propiedad de insertar código del ID



Creamos el constructor de la clase MyDate



Cambiamos código de la case

The screenshot shows the NetBeans IDE interface with the title bar "Date - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar has icons for creating files, opening, saving, and running. The Projects tab shows a project named "Date" containing "Web Pages" (with "index.html"), "Source Packages" (containing "co.com.date" with "MyDate.java"), "Test Packages", "Libraries" (with "JDK 1.8 (Default)" and "GlassFish Server 4.1.1"), "Test Libraries", and "Configuration Files" (with "MANIFEST.MF"). The Files tab is selected, showing the "MyDate.java" file. The code in "MyDate.java" is:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package co.com.date;

import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;

/**
 *
 * @author LeonardoDesarrollo
 */
public class MyDate {
    public Date today;
    public DateFormat dateFormat;

    public MyDate() {
        today = new Date();
    }

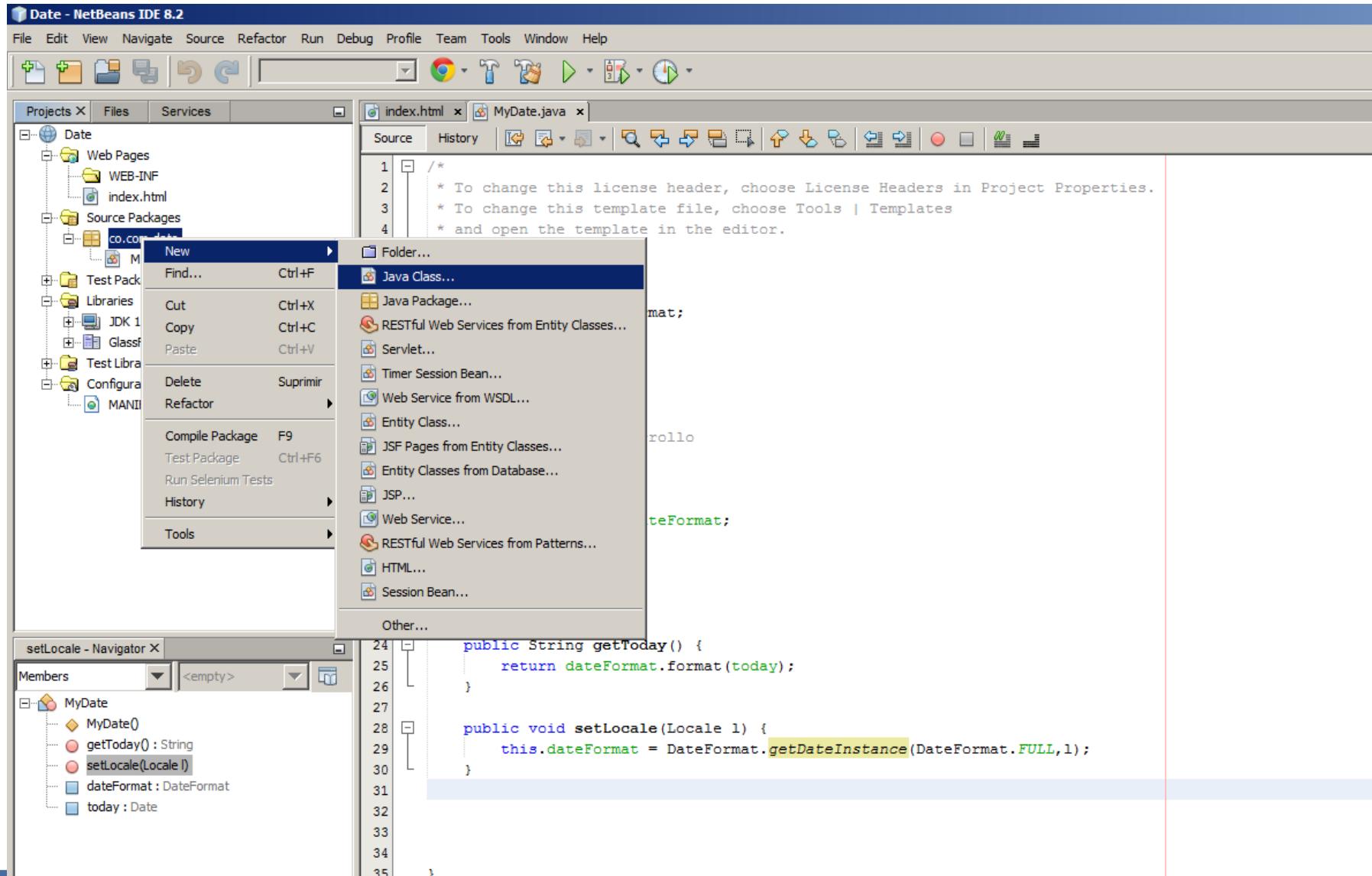
    public String getToday() {
        return dateFormat.format(today);
    }

    public void setLocale(Locale l) {
        this.dateFormat = DateFormat.getDateInstance(DateFormat.FULL, l);
    }
}
```

The Navigator tab at the bottom left shows the members of the "MyDate" class: MyDate(), getToday():String, setLocale(Locale l), dateFormat:DateFormat, and today:Date.



Creamos una nueva clase



Adicionamos código a la clase MyLocales

The screenshot shows the NetBeans IDE 8.2 interface with the following details:

- Project Explorer (Left):** Shows the **Date** project structure:
 - Web Pages: index.html
 - Source Packages: co.com.date (containing MyDate.java and MyLocales.java)
 - Test Packages
 - Libraries: JDK 1.8 (Default), GlassFish Server 4.1.1
 - Test Libraries
 - Configuration Files: MANIFEST.MF
- Code Editor (Center):** Displays the **MyLocales.java** file content:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package co.com.date;

import java.text.DateFormat;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.HashMap;
import java.util.Locale;

/**
 *
 * @author LeonardoDesarrollo
 */

public class MyLocales {
    public ArrayList<String> localeNames;
    public HashMap <String, Locale> locales;
    public Locale selectedLocale;
    public String selectedLocaleString;

    public MyLocales() {
        locales = new HashMap<String, Locale>();
        localeNames = new ArrayList<>();

        Locale[] list = DateFormat.getAvailableLocales();

        for (int i = 0; i < list.length; i++) {
            locales.put(list[i].getDisplayName(), list[i]);
            localeNames.add(list[i].getDisplayName());
        }
        Collections.sort(localeNames);
        selectedLocale = null;
        selectedLocaleString = null;
    }
}
```
- Navigator (Bottom Left):** Shows the members of the **MyLocales** class:
 - Members
 - MyLocales
 - MyLocales()
 - localeNames : ArrayList<String>
 - locales : HashMap <String, Locale>
 - selectedLocale : Locale
 - selectedLocaleString : String

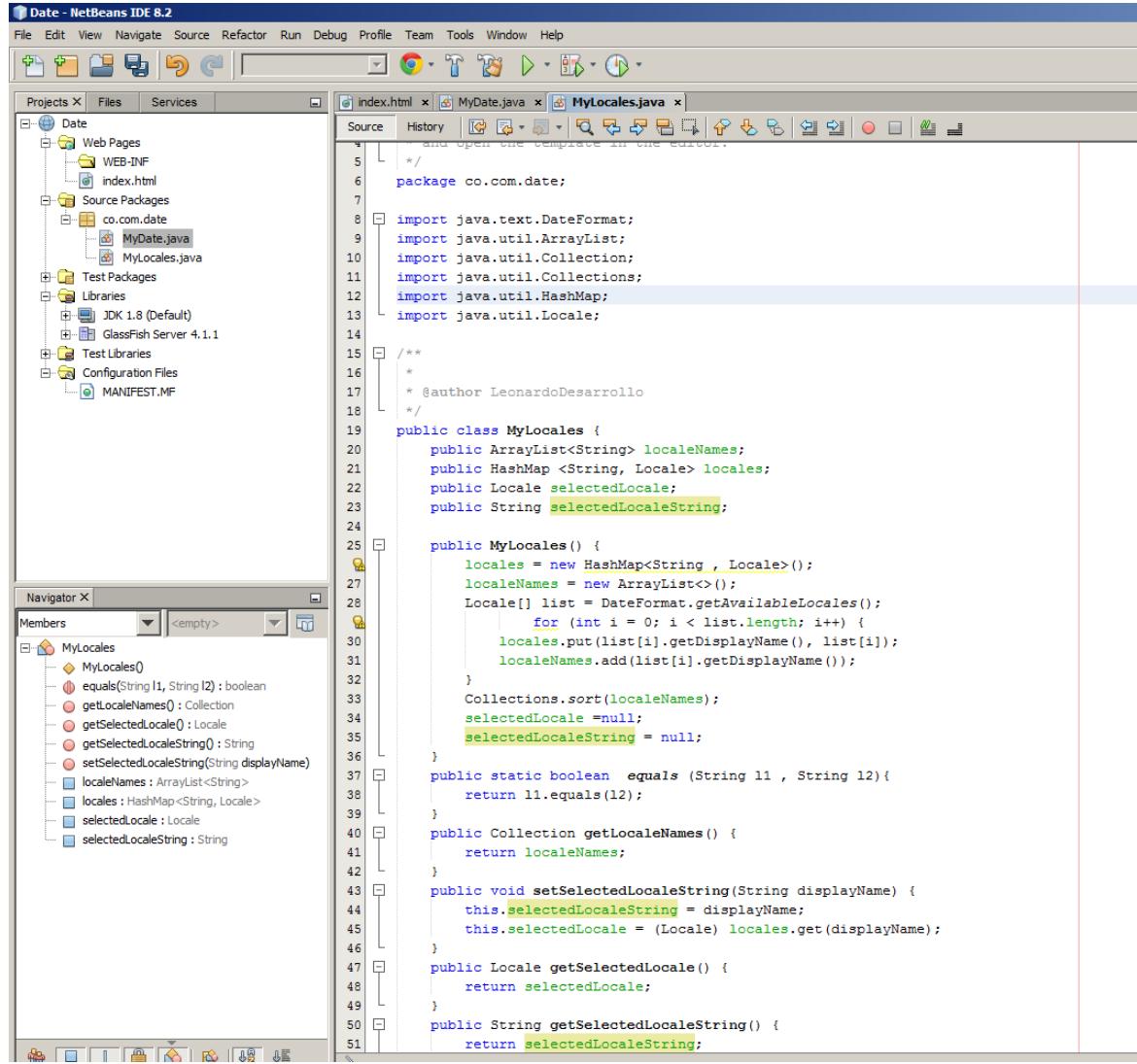


Adicionamos código a la clase MyLocales

```
public class MyLocales {
    public ArrayList<String> localeNames;
    public HashMap <String, Locale> locales;
    public Locale selectedLocale;
    public String selectedLocaleString;
    public MyLocales() {
        locales = new HashMap<String , Locale>();
        localeNames = new ArrayList<>();
        Locale[] list = DateFormat.getAvailableLocales();
        for (int i = 0; i < list.length; i++) {
            locales.put(list[i].getDisplayName(), list[i]);
            localeNames.add(list[i].getDisplayName());
        }
        Collections.sort(localeNames);
        selectedLocale = null;
        selectedLocaleString = null;
    }
    public static boolean equals (String l1 , String l2){
        return l1.equals(l2);
    }
    public Collection getLocaleNames () {
        return localeNames;
    }
    public void setSelectedLocaleString (String displayName) {
        this.selectedLocaleString = displayName;
        this.selectedLocale = (Locale) locales.get(displayName);
    }
    public Locale getSelectedLocale () {
        return selectedLocale;
    }
    public String getSelectedLocaleString () {
        return selectedLocaleString;
    }
}
```



Adicionamos código a la clase MyLocales

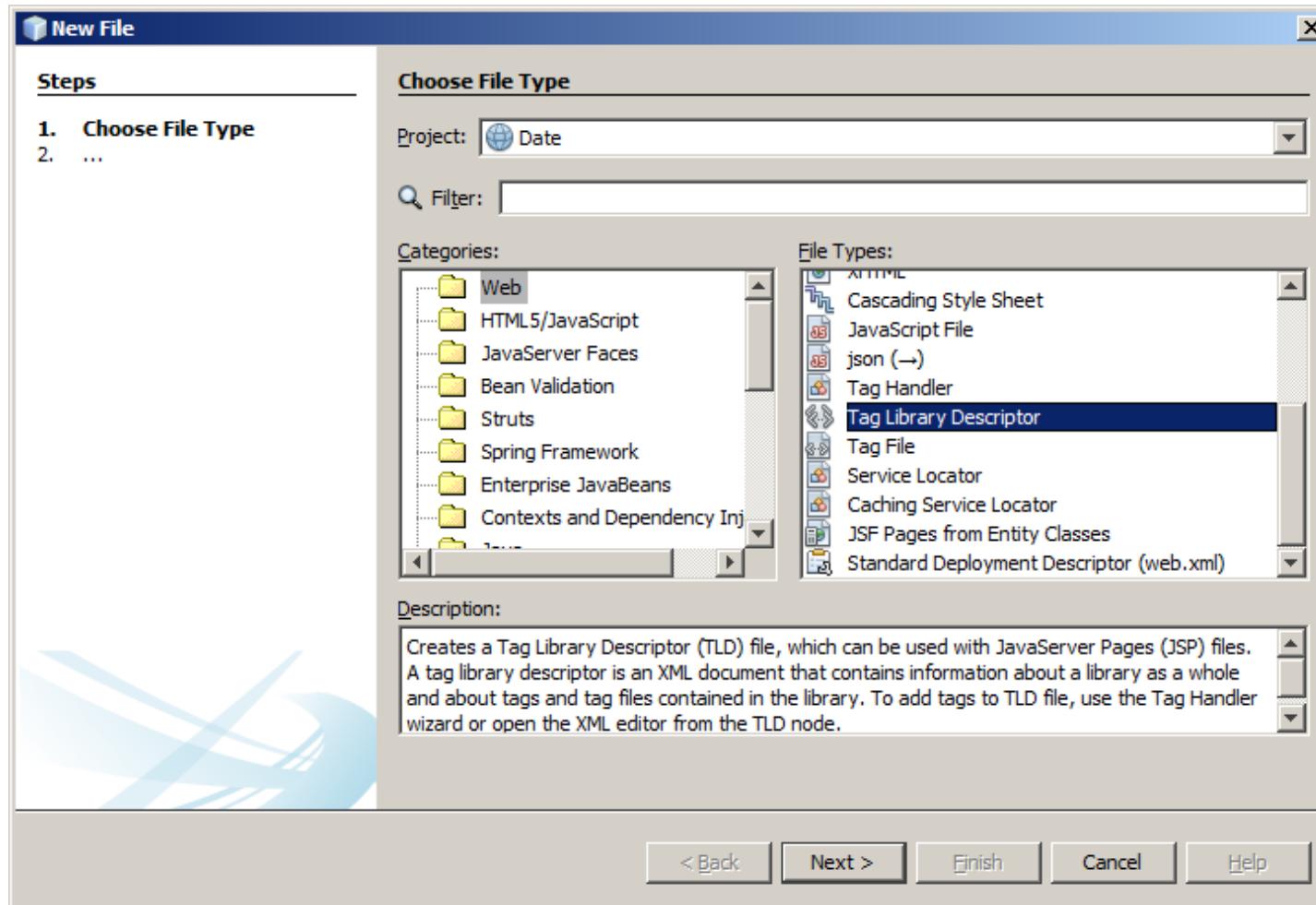


The screenshot shows the NetBeans IDE 8.2 interface with the following details:

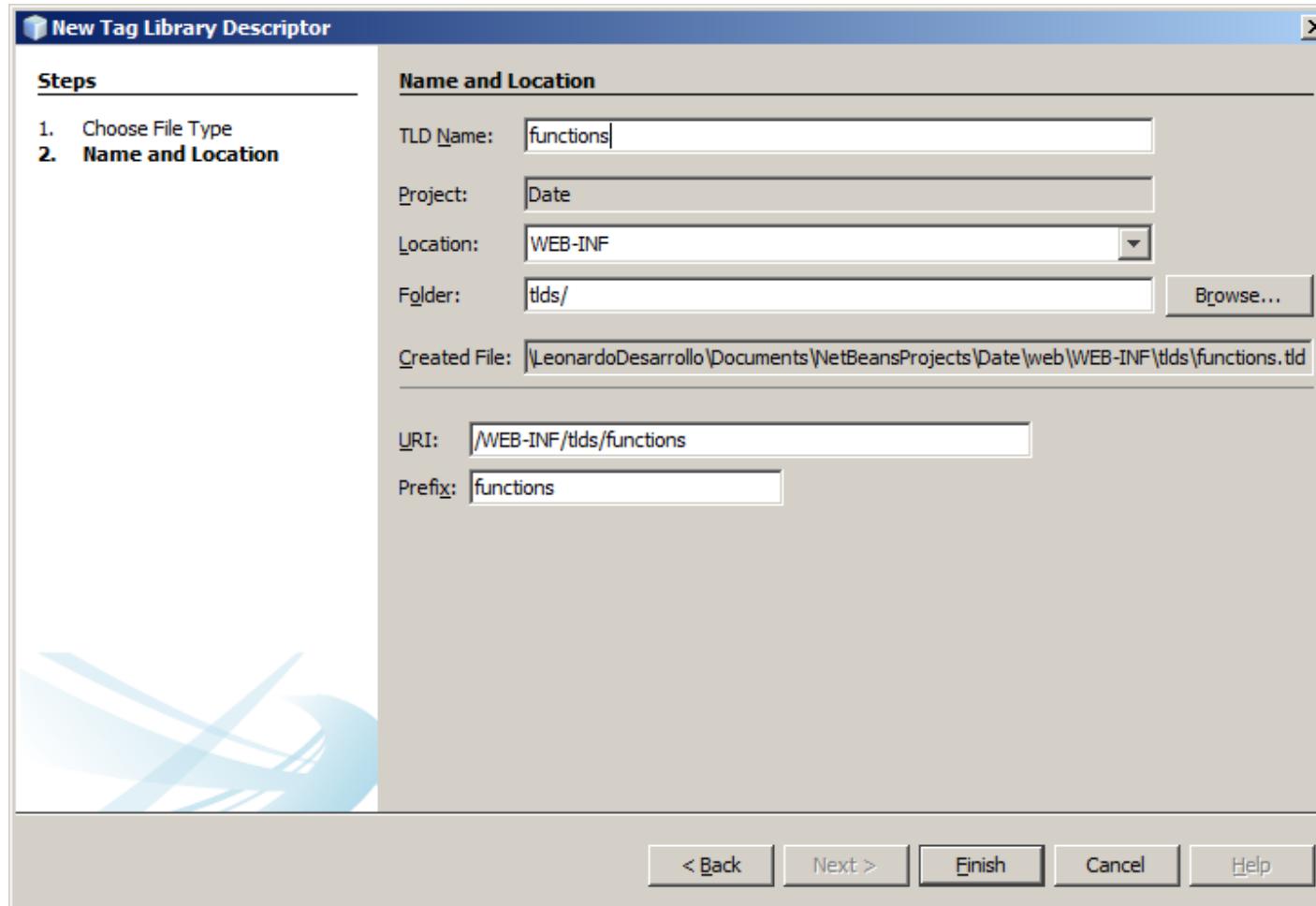
- Title Bar:** Date - NetBeans IDE 8.2
- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Toolbar:** Standard NetBeans toolbar icons.
- Project Explorer:** Shows a project named "Date" containing "Web Pages" (index.html), "Source Packages" (co.com.date package with MyDate.java and MyLocales.java), "Test Packages", "Libraries" (JDK 1.8 (Default), GlassFish Server 4.1.1), "Test Libraries", and "Configuration Files" (MANIFEST.MF).
- Code Editor:** The "MyLocales.java" file is open, displaying Java code for managing locales. The code includes imports for java.util.ArrayList, java.util.Collection, java.util.Collections, java.util.HashMap, and java.util.Locale. It defines a class MyLocales with methods for locale names, selected locale, and equality checks.
- Navigator:** Shows the members of the MyLocales class, including its constructor, equals, getLocaleNames, getSelectedLocale, setSelectedLocaleString, and several private fields and methods.
- Status Bar:** Standard NetBeans status bar.



Creamos un TLD



Creamos un TLD



Creamos un TLD

The screenshot shows the NetBeans IDE 8.2 interface with the following details:

- Title Bar:** Date - NetBeans IDE 8.2
- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Toolbar:** Standard NetBeans toolbar with icons for new project, open project, save, run, etc.
- Project Explorer (Projects X):** Shows a project named "Date" containing "Web Pages", "Source Packages", "Test Packages", "Libraries", "Test Libraries", and "Configuration Files". Inside "Web Pages", there is a "WEB-INF" folder with "tlds" and "functions.tld".
- Code Editor (MyLocales.java X):** Shows Java code for a class named MyLocales.java.
- Code Editor (functions.tld X):** The active editor displays the XML configuration for a Tag Library. The XML content includes:

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      tlib-version="1.0" short-name="functions"
      uri="/WEB-INF/tlds/functions">

    <!-- A validator verifies that the tags are used correctly at JSP
         translation time. Validator entries look like this:
    <validator>
        <validator-class>com.mycompany.TagLibValidator</validator-class>
        <init-param>
            <param-name>parameter</param-name>
            <param-value>value</param-value>
        </init-param>
    </validator>
    -->
    <!-- A tag library can register Servlet Context event listeners in
         case it needs to react to such events. Listener entries look
         like this:
    <listener>
        <listener-class>com.mycompany.TagLibListener</listener-class>
    </listener>
    -->
    <function>
        <name>equals</name>
        <function-class>co.com.date.MyLocales</function-class>
        <function-signature>boolean equals (java.lang.String ,java.lang.String)</function-signature>
    </function>
</taglib>
```
- Navigator X:** Shows the structure of an HTML page with sections like html, head, meta, title, body, and h1.



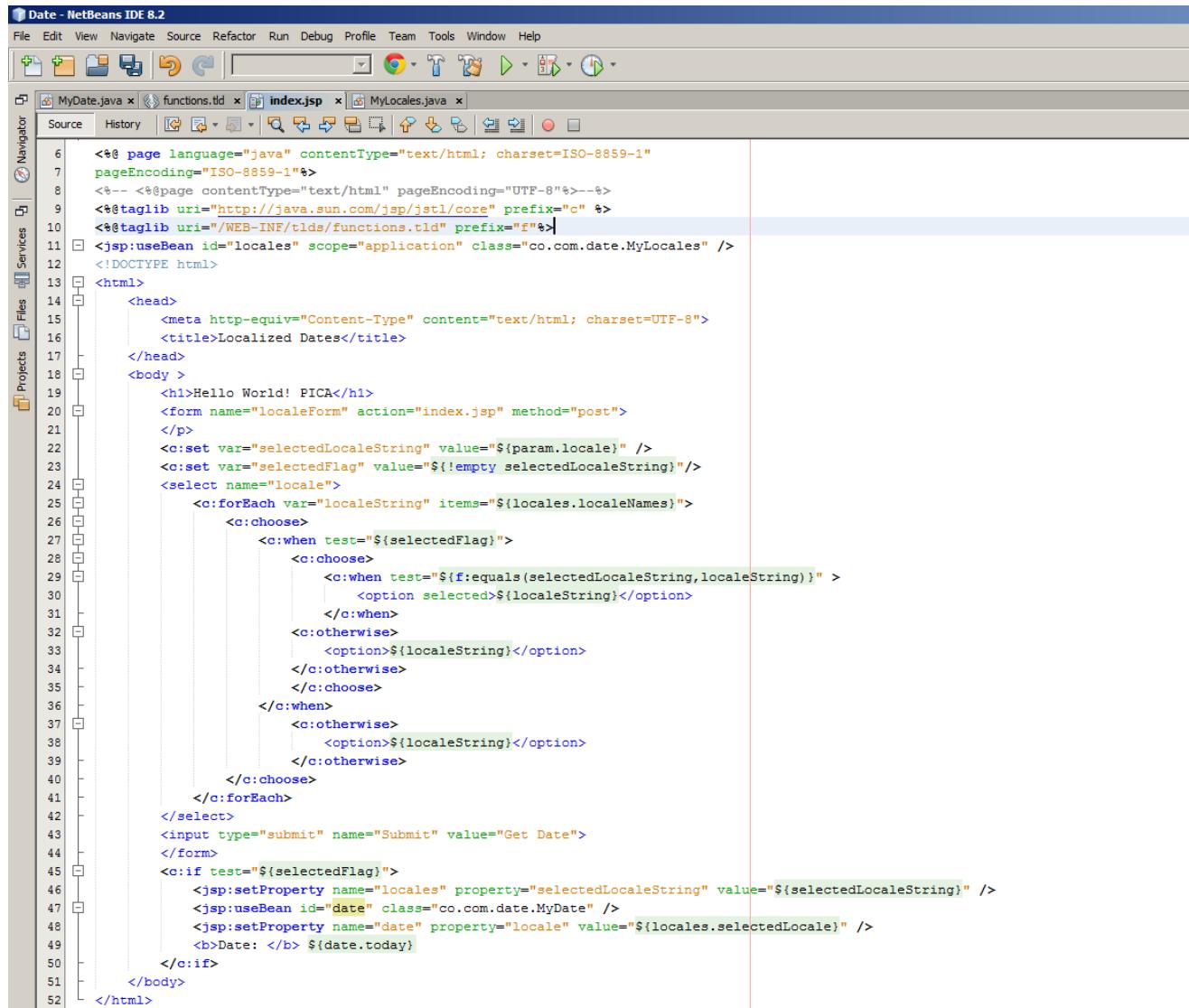
Creamos una JSP

The screenshot shows the NetBeans IDE 8.2 interface with the following details:

- Title Bar:** Date - NetBeans IDE 8.2
- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Toolbar:** Standard NetBeans toolbar with icons for New Project, Open Project, Save, Undo, Redo, Find, Replace, Run, Stop, etc.
- Project Explorer:** Shows a project named "Date" containing:
 - Web Pages: WEB-INF/tlds/functions.tld, index.jsp
 - Source Packages: co.com.date (MyDate.java, MyLocales.java)
 - Test Packages
 - Libraries: JDK 1.8 (Default), GlassFish Server 4.1.1
 - Test Libraries
 - Configuration Files: MANIFEST.MF
- Code Editor:** The "index.jsp" file is open in the Source tab. The code is as follows:

```
<%--  
    Document      : index  
    Created on   : 29/01/2017, 07:02:53 PM  
    Author       : LeonardoDesarrollo  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  
<%@taglib uri="/WEB-INF/tlds/functions.tld" prefix="f"%>  
<!DOCTYPE html>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
        <title>Localized Dates</title>  
    </head>  
    <body>  
        <h1>Hello World!</h1>  
        <jsp:useBean id="locales" scope="application" class="co.com.date.MyLocales" />  
    </body>  
</html>
```
- Navigator:** Shows the outline of the "index.jsp" file structure:
 - html
 - head
 - meta
 - title
 - body
 - h1

Modificamos el código de la JSP



The screenshot shows the NetBeans IDE 8.2 interface with the title bar "Date - NetBeans IDE 8.2". The code editor displays JSP code for a page named "index.jsp". The code includes Java scriptlets, JSTL tags, and custom tags from "MyLocales.java". The code is responsible for displaying localized dates based on user selection. The code editor has syntax highlighting and a vertical red margin line.

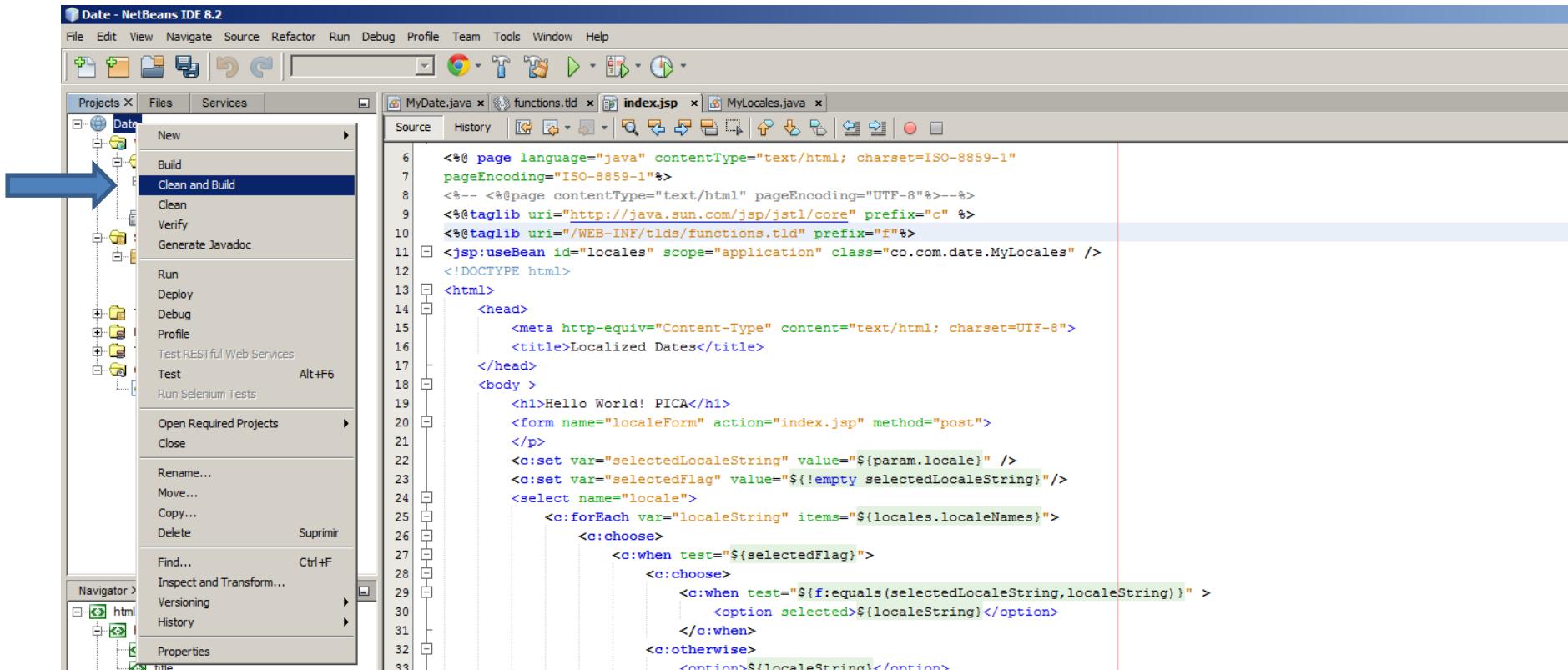
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%-- <%@page contentType="text/html" pageEncoding="UTF-8"%>--%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="/WEB-INF/tlds/functions.tld" prefix="f" %>
<jsp:useBean id="locales" scope="application" class="co.com.date.MyLocales" />
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Localized Dates</title>
    </head>
    <body>
        <h1>Hello World! PICA</h1>
        <form name="localeForm" action="index.jsp" method="post">
        </p>
        <c:set var="selectedLocaleString" value="${param.locale}" />
        <c:set var="selectedFlag" value="${!empty selectedLocaleString}" />
        <select name="locale">
            <c:forEach var="localeString" items="${locales.localeNames}">
                <c:choose>
                    <c:when test="${selectedFlag}">
                        <c:choose>
                            <c:when test="${f>equals(selectedLocaleString,localeString)}" >
                                <option selected>${localeString}</option>
                            </c:when>
                            <c:otherwise>
                                <option>${localeString}</option>
                            </c:otherwise>
                        </c:choose>
                    </c:when>
                    <c:otherwise>
                        <option>${localeString}</option>
                    </c:otherwise>
                </c:choose>
            </c:forEach>
        </select>
        <input type="submit" name="Submit" value="Get Date">
    </form>
    <c:if test="${selectedFlag}">
        <jsp:setProperty name="locales" property="selectedLocaleString" value="${selectedLocaleString}" />
        <jsp:useBean id="date" class="co.com.date.MyDate" />
        <jsp:setProperty name="date" property="locale" value="${locales.selectedLocale}" />
        <b>Date: </b> ${date.today}
    </c:if>
    </body>
</html>
```



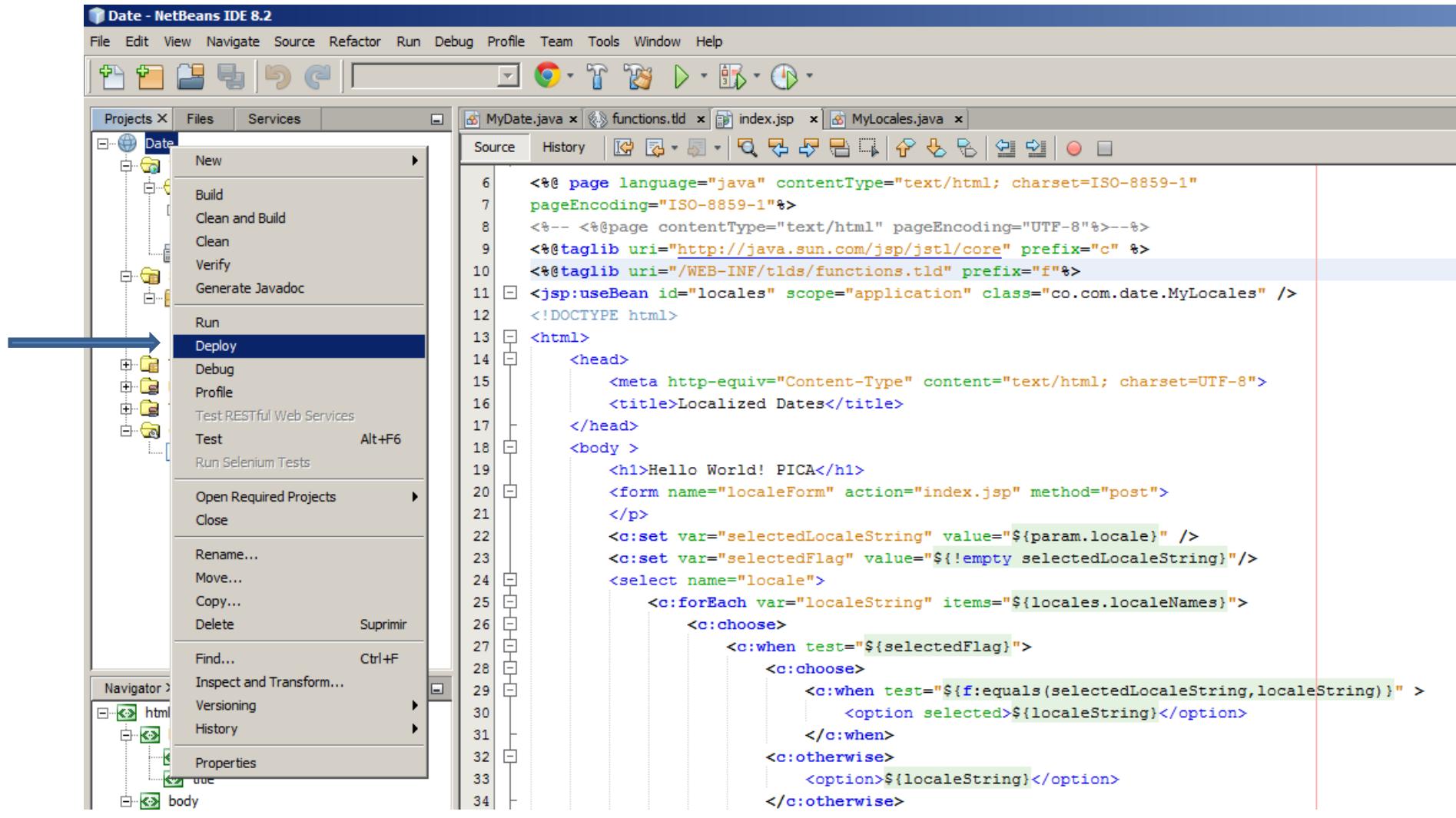
Modificamos el código de la JSP

```
<jsp:useBean id="locales" scope="application" class="co.com.date.MyLocales" />
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Localized Dates</title>
    </head>
    <body>
        <h1>Hello World! PICA</h1>
        <form name="localeForm" action="index.jsp" method="post">
            <p>
                <c:set var="selectedLocaleString" value="${param.locale}" />
                <c:set var="selectedFlag" value="${!empty selectedLocaleString}" />
                <select name="locale">
                    <c:forEach var="localeString" items="${locales.localeNames}">
                        <c:choose>
                            <c:when test="${selectedFlag}">
                                <c:choose>
                                    <c:when test="${f>equals(selectedLocaleString,localeString)}" >
                                        <option selected>${localeString}</option>
                                    </c:when>
                                    <c:otherwise>
                                        <option>${localeString}</option>
                                    </c:otherwise>
                                </c:choose>
                            </c:when>
                            <c:otherwise>
                                <option>${localeString}</option>
                            </c:otherwise>
                        </c:choose>
                    </c:forEach>
                </select>
                <input type="submit" name="Submit" value="Get Date" />
            </form>
            <c:if test="${selectedFlag}">
                <jsp:setProperty name="locales" property="selectedLocaleString" value="${selectedLocaleString}" />
                <jsp:useBean id="date" class="co.com.date.MyDate" />
                <jsp:setProperty name="date" property="locale" value="${locales.selectedLocale}" />
                <b>Date: </b> ${date.today}
            </c:if>
        </body>
    </html>
```

Preparamos para compilar el proyecto



Desplegamos el proyecto



Probamos el proyecto



Hello World! PICA

▼



Hello World! PICA

▼

- albanés
- albanés (Albania)
- alemán
- alemán (Alemania)
- alemán (Austria)
- alemán (Grecia)
- alemán (Luxemburgo)
- alemán (Suiza)
- bielorruso
- bielorruso (Bielorrusia)
- búlgaro
- búlgaro (Bulgaria)
- catalán
- catalán (España)
- checho**
- checho (Chequia)
- chino
- chino (China)
- chino (Hong Kong)



Hello World! PICA

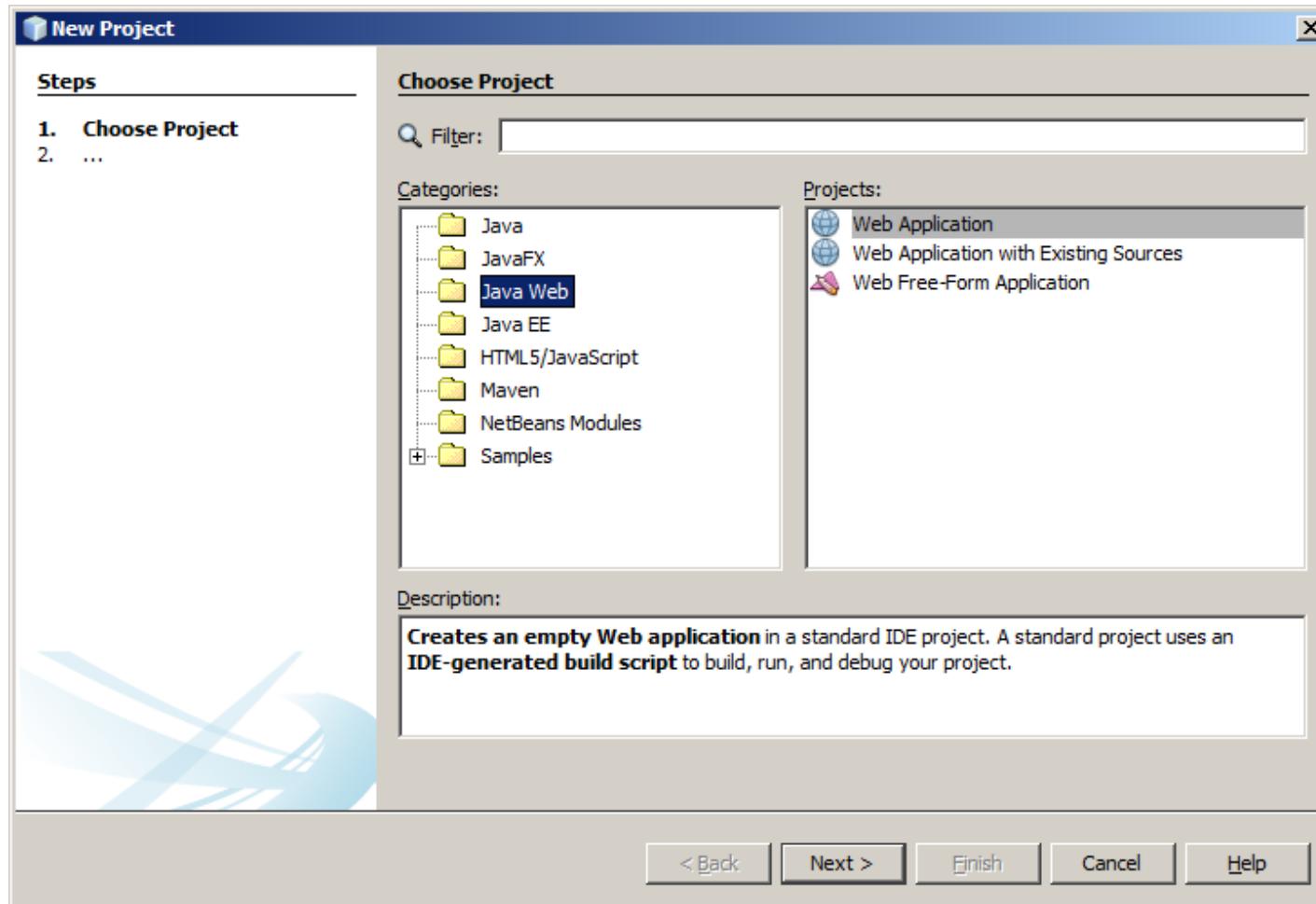
▼

Date: diumenge, 29 / de gener / 2017

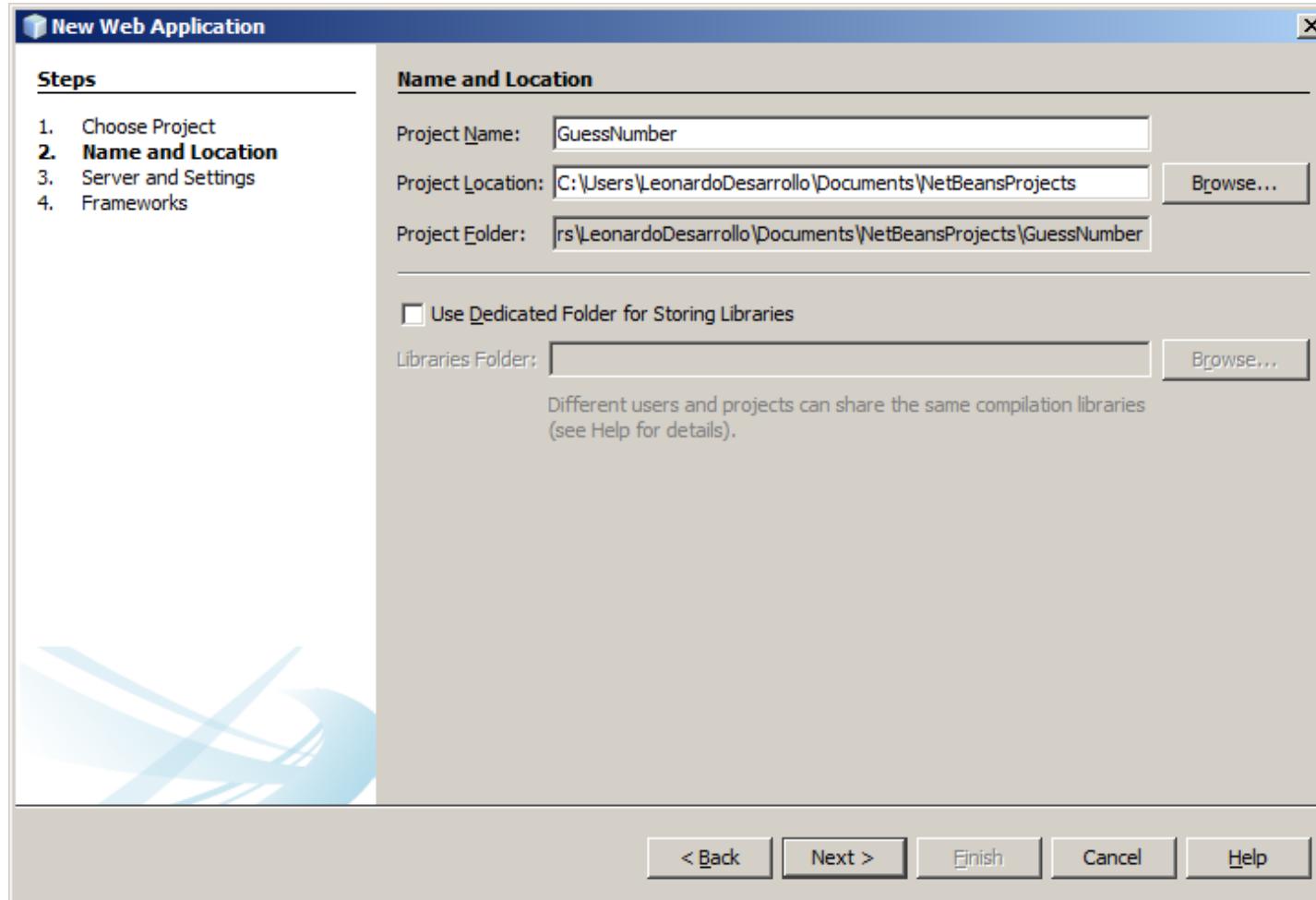
JSF proyecto GuessNumber.

Consiste en adivinar un número que se genera aleatoriamente a partir de una función Random.
El rango de los números va de 0 a 10

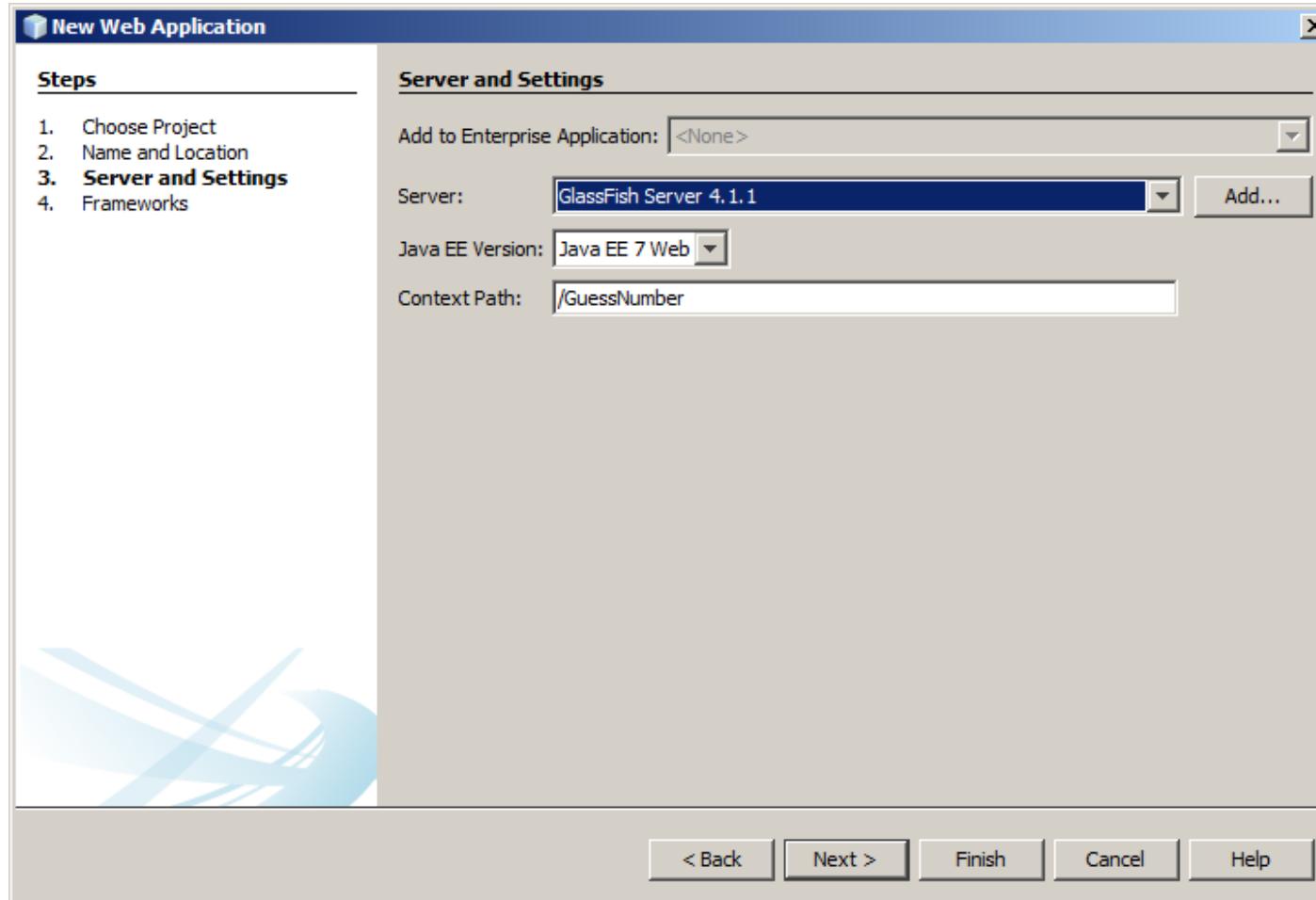
Creamos un nuevo proyecto Web



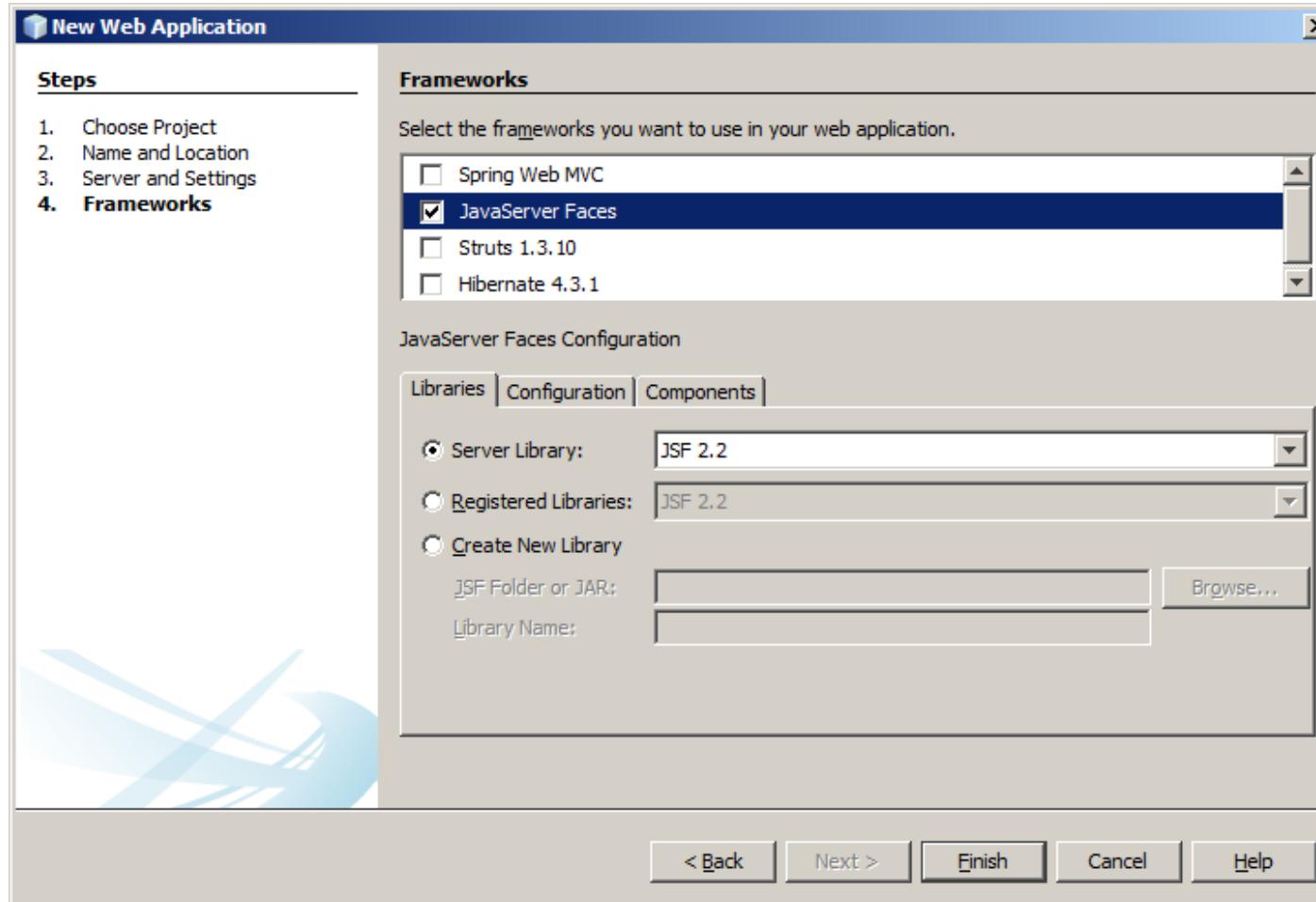
Creamos un nuevo proyecto Web



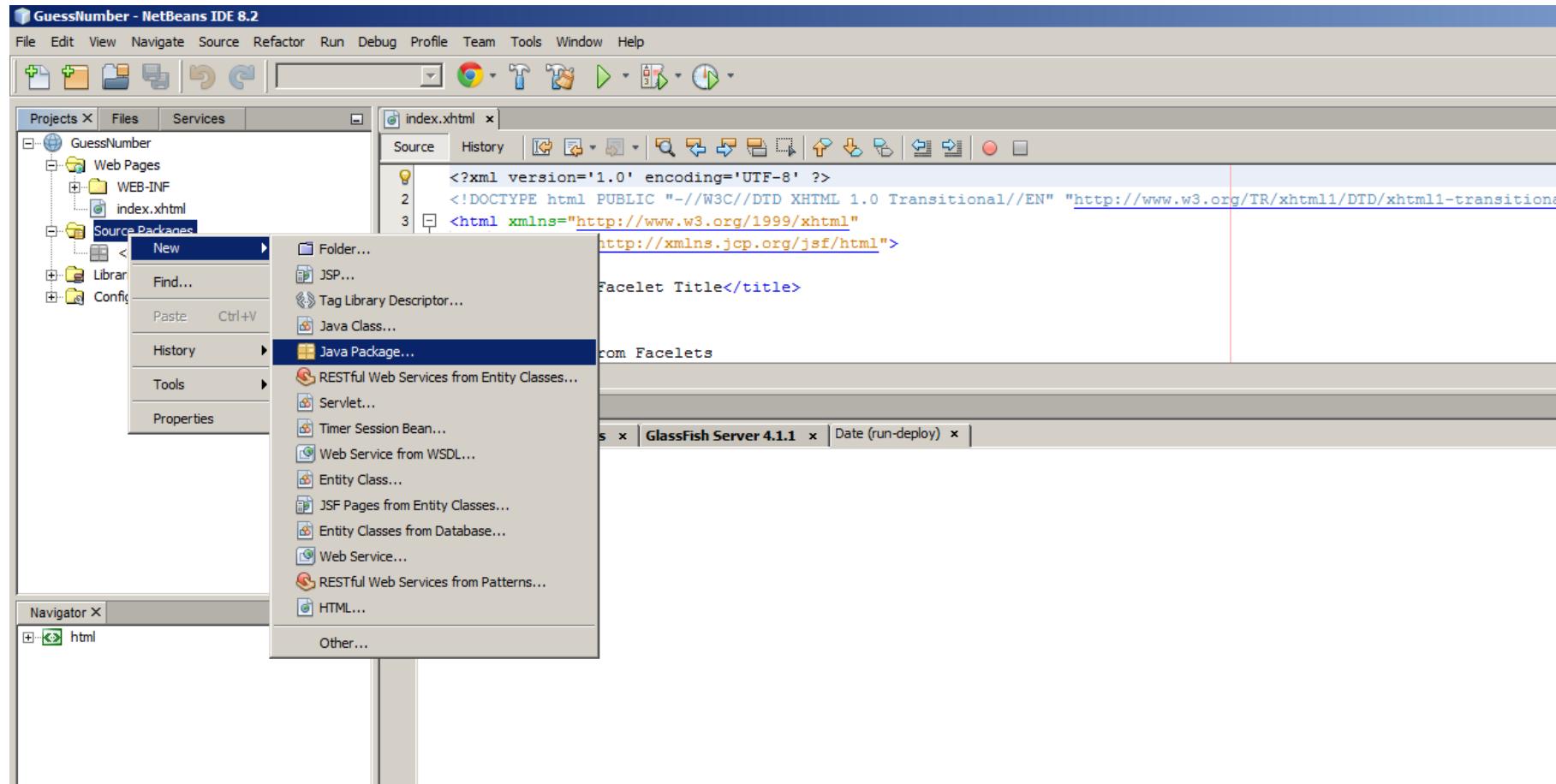
Configuramos el servidor



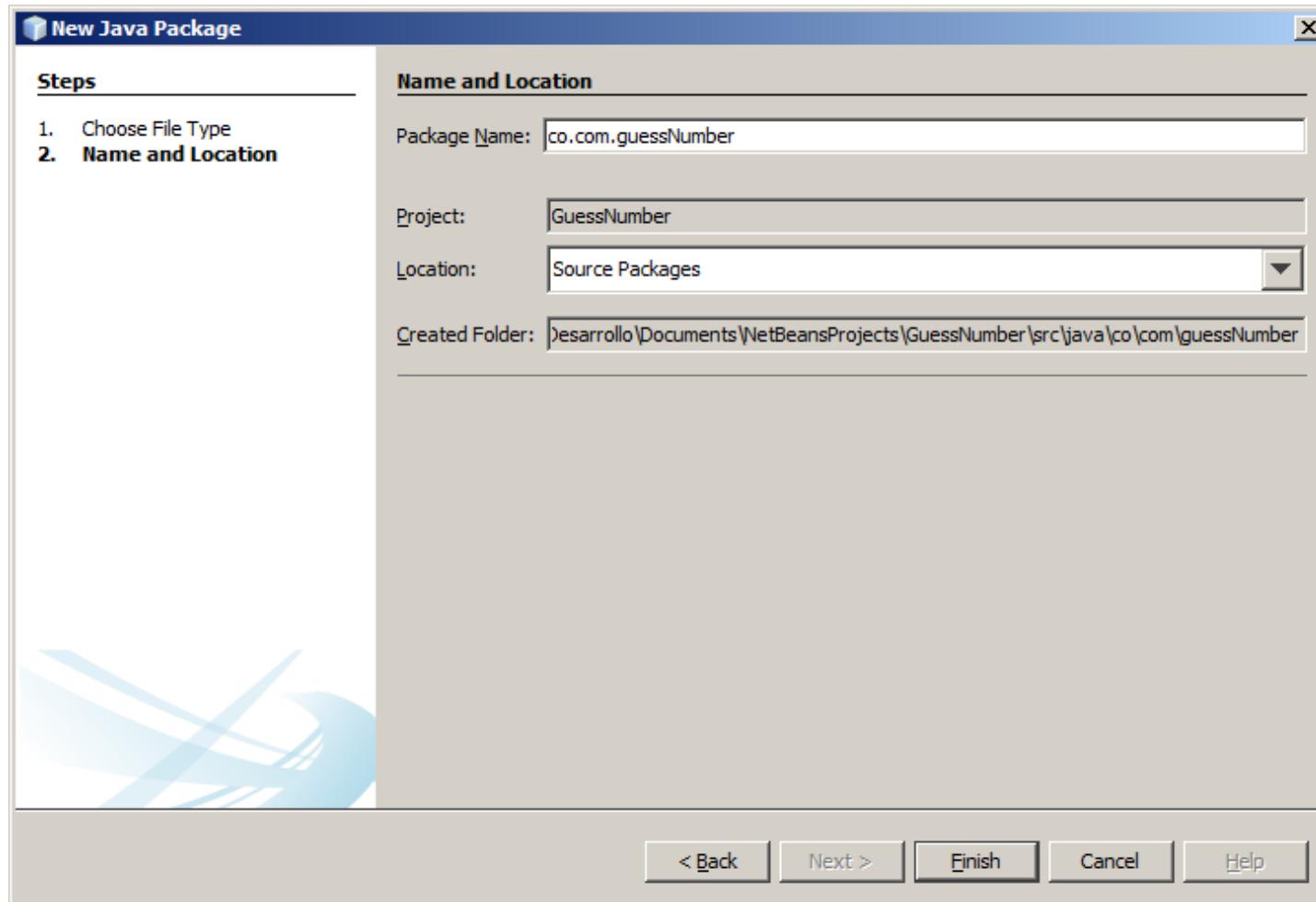
Seleccionamos el framework JavaServer Faces



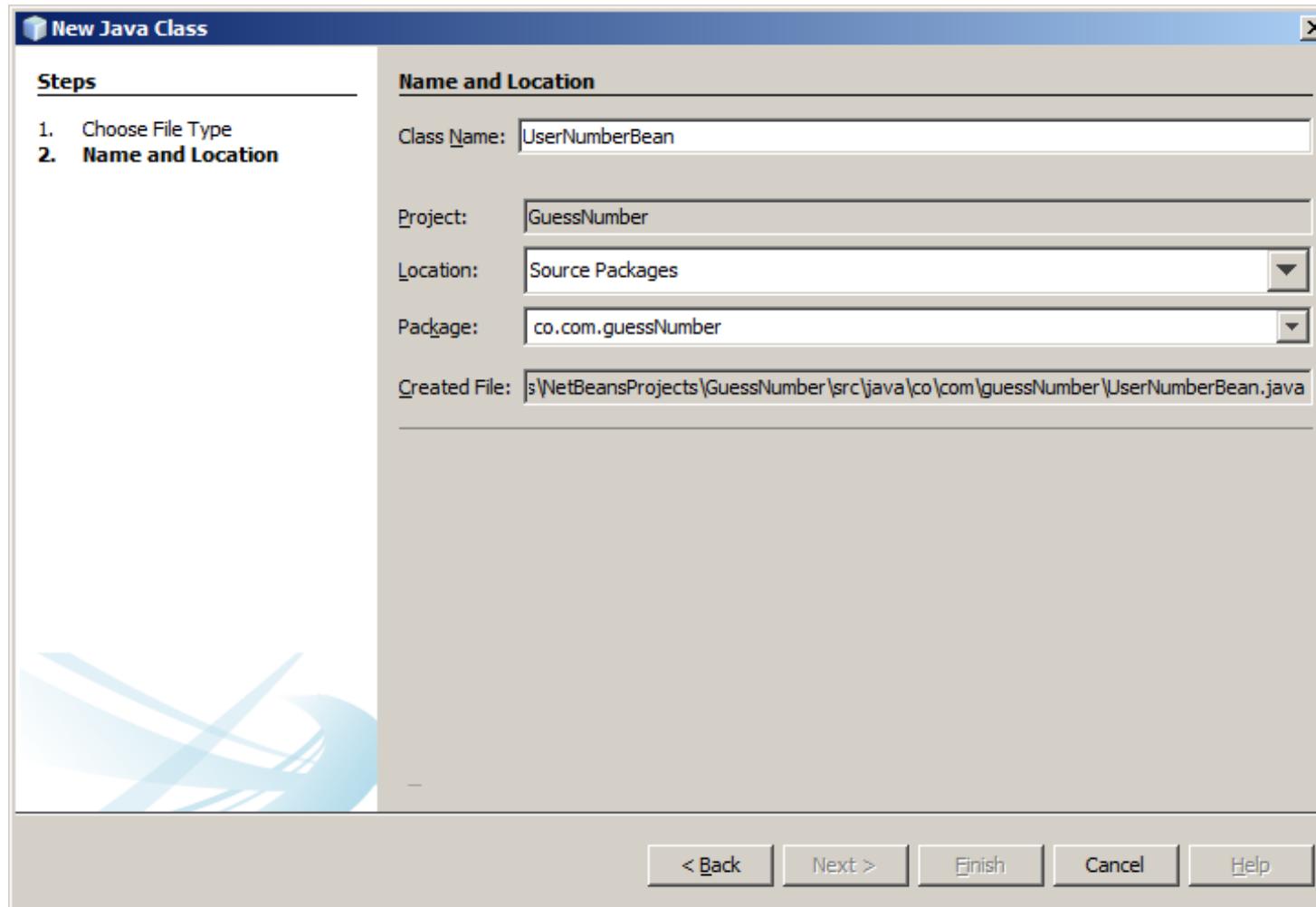
Creamos un nuevo paquete para las clases



Creamos un nuevo paquete para las clases



Creamos una nueva clase GuessNumber



Adicionamos el código a la clase GuessNumber

The screenshot shows the NetBeans IDE 8.2 interface with the following details:

- Title Bar:** GuessNumber - NetBeans IDE 8.2
- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Toolbar:** Standard NetBeans toolbar with icons for file operations.
- Project Explorer (Projects X):** Shows the project structure: GuessNumber > Web Pages > index.xhtml; Source Packages > co.com.guessNumber > UserNumberBean.java; Test Packages, Libraries, Test Libraries, Configuration Files.
- Code Editor (UserNumberBean.java):** Displays the Java code for the UserNumberBean class. The code includes a license header, package declaration, author annotation, and class definition with fields: randomInt, userNumber, responce, maximum, and minimum.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package co.com.guessNumber;

/**
 *
 * @author LeonardoDesarrollo
 */
public class UserNumberBean {
    public Integer randomInt = null;
    public Integer userNumber = null;
    public String responce = null;
    private double maximum=0;
    private double minimum=0;
}
```

- Navigator (Navigator X):** Shows the members of the UserNumberBean class: maximum, minimum, randomInt, responce, and userNumber.



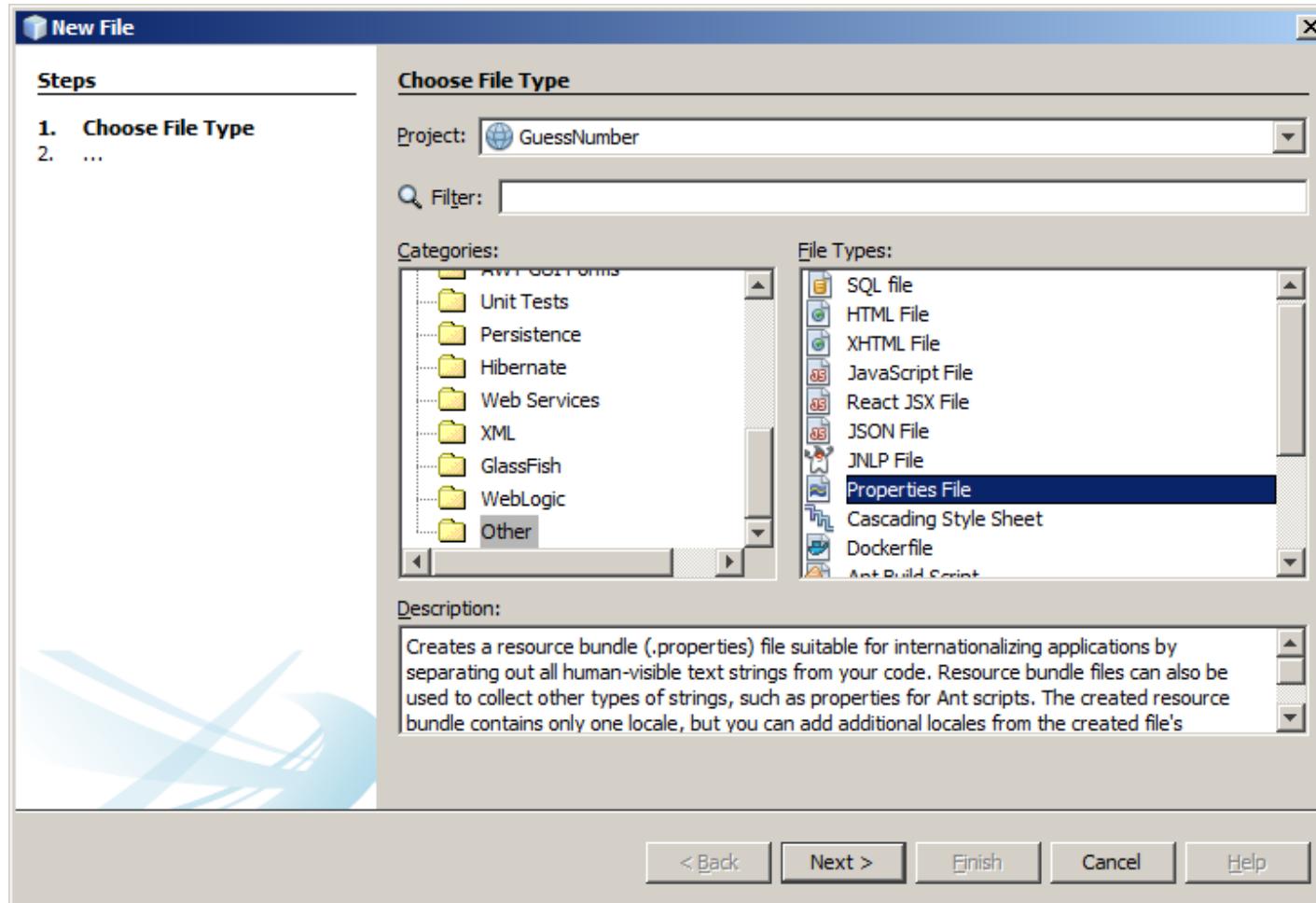
Adicionamos el código, constructor y los gets y sets

The screenshot shows the NetBeans IDE 8.2 interface with the following details:

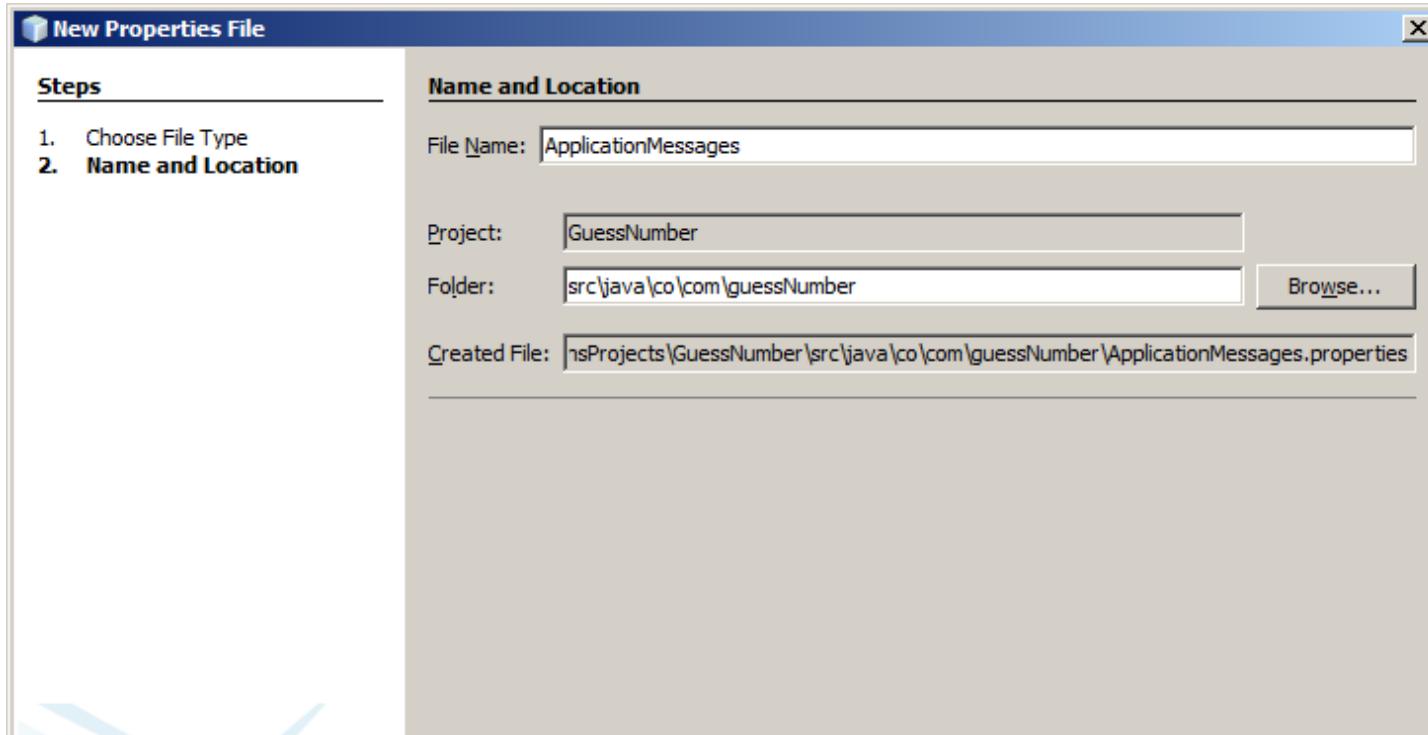
- Project Structure:** The "Projects" tab shows a project named "GuessNumber" containing "Web Pages", "Source Packages" (with "co.com.guessNumber" selected), and "Test Packages".
- Code Editor:** The "UserNumberBean.java" file is open, displaying Java code for a bean. The code includes fields for randomInt, userNumber, response, maximum, and minimum, along with their corresponding getters and setters.
- Context Menu:** A context menu is open over the "minimum" field, specifically at line 20. The menu is titled "Generate" and lists options such as "Constructor...", "getRandomInt()", "Logger...", "equals() and hashCode()", "toString()", "Delegate Method...", "Override Method...", "Add Property...", "Call Enterprise Bean...", "Use Database...", and "Send JMS Message...".
- Navigator:** The "minimum - Navigator X" window shows the members of the "UserNumberBean" class, including the generated getters and setters for maximum, minimum, getRandomInt, response, and userNumber.



Creamos un archivo de propiedades



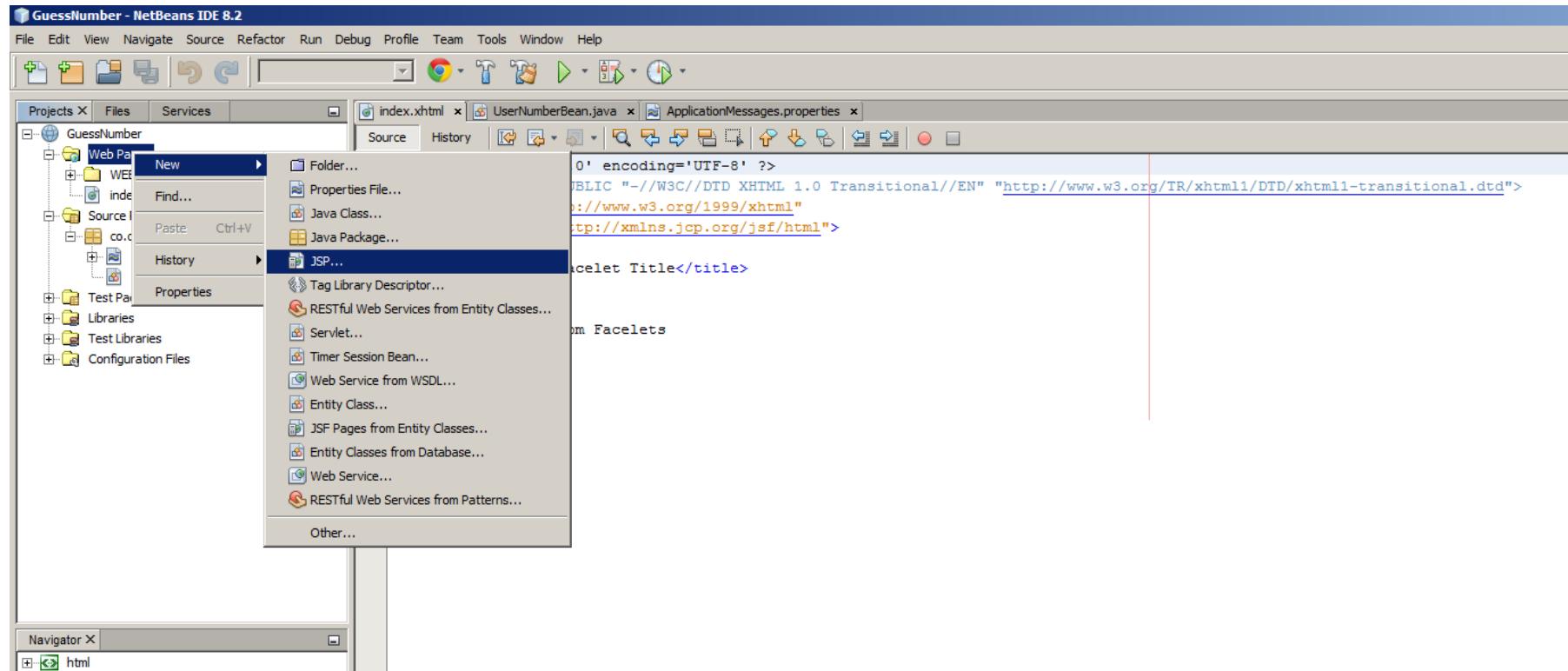
Creamos un archivo de propiedades



Asignamos la propiedad :

userNoConvert=El valor ingresado no es un numero.

Creamos una nueva JSP



Creamos una nueva JSP y lo llamamos response

GuessNumber - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects X Files Services

GuessNumber

- Web Pages
 - WEB-INF
 - Responce.jsp
 - index.xhtml
 - Source Packages
 - co.com.guessNumber
 - ApplicationMessages.properties
 - UserNumberBean.java
 - Test Packages
 - Libraries
 - Test Libraries
 - Configuration Files

UserNumberBean.java ApplicationMessages.properties index.xhtml Responce.jsp

Source History

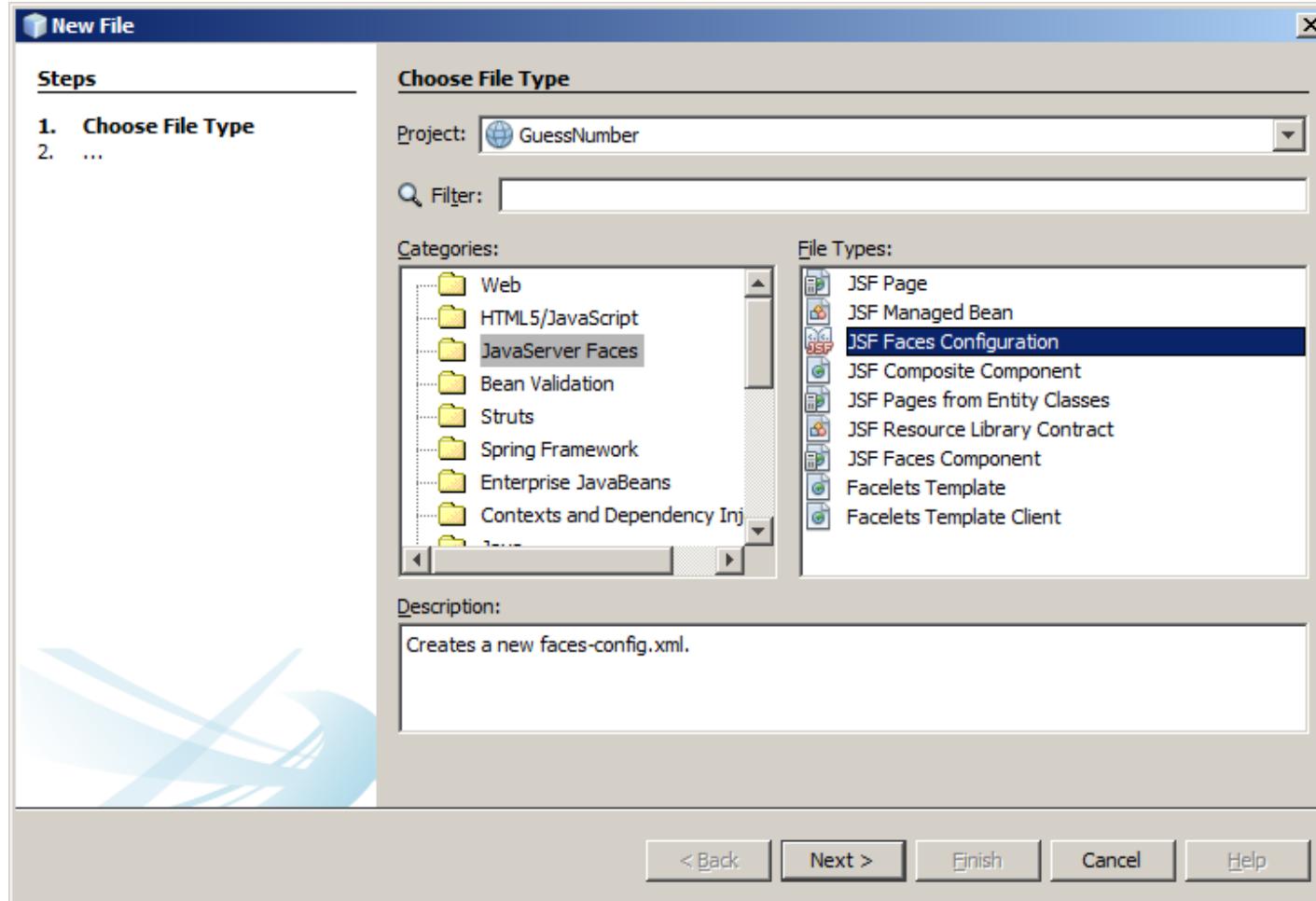
```
<%--  
Document : Responce  
Created on : 29/01/2017, 11:24:48 PM  
Author : LeonardoDesarrollo  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
    <head> <title>Guess The Number</title> </head>  
    <%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>  
    <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>  
    <body bgcolor="white">  
        <f:view>  
            <h:form id="responseForm" >  
                <h:graphicImage id="waveImg" url="/wave.med.gif" />  
                <h2><h:outputText id="result"  
                    value="#{UserNumberBean.response}" /></h2>  
                <h:commandButton id="back" value="Back" action="success"/><p>  
            </h:form>  
        </f:view>  
    </body>  
</html>
```

Navigator X

 - html
 - head
 - title
 - body
 - h2



Creamos un nuevo archivo de configuración JSF



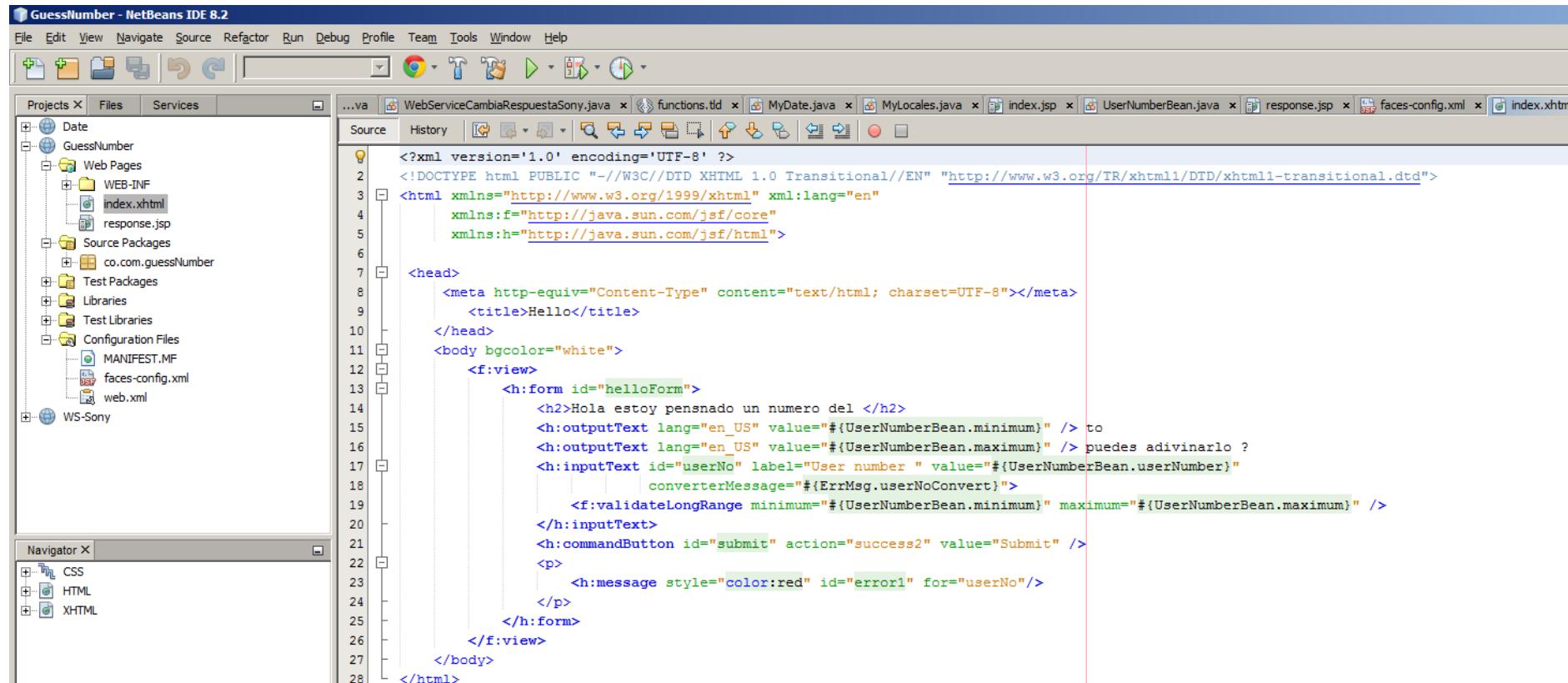
Creamos un nuevo archivo de configuración JSF

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="2.2"
  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-
facesconfig_2_2.xsd">
  <application>
    <resource-bundle>
      <base-name>co.com.guessNumber.ApplicationMessages</base-name>
      <var>ErrMsg</var>
    </resource-bundle>
    <locale-config>
      <default-locale>en</default-locale>
      <supported-locale>de</supported-locale>
      <supported-locale>fr</supported-locale>
      <supported-locale>es</supported-locale>
    </locale-config>
  </application>
  <navigation-rule>
    <description>
    </description>
    <from-view-id>/index.xhtml</from-view-id>
    <navigation-case>
      <description>
      </description>
      <from-outcome>success2</from-outcome>
      <to-view-id>/response.jsp</to-view-id>
    </navigation-case>
  </navigation-rule>
  <navigation-rule>
    <description>
    </description>
    <from-view-id>/response.jsp</from-view-id>
    <navigation-case>
      <description>
      </description>
      <from-outcome>success</from-outcome>
      <to-view-id>/index.xhtml</to-view-id>
    </navigation-case>
  </navigation-rule>
  <managed-bean>
    <description>
    </description>
    <managed-bean-name>UserNumberBean</managed-bean-name>
    <managed-bean-class>co.com.guessNumber.UserNumberBean</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
    <managed-property>
      <property-name>minimum</property-name>
      <property-class>double</property-class>
      <value>0</value>
    </managed-property>
    <managed-property>
      <property-name>maximum</property-name>
      <property-class>double</property-class>
      <value>10</value>
    </managed-property>
  </managed-bean>
</faces-config>
```

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="2.2"
  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-
facesconfig_2_2.xsd">
  <application>
    <resource-bundle>
      <base-name>co.com.guessNumber.ApplicationMessages</base-name>
      <var>ErrMsg</var>
    </resource-bundle>
    <locale-config>
      <default-locale>en</default-locale>
      <supported-locale>de</supported-locale>
      <supported-locale>fr</supported-locale>
      <supported-locale>es</supported-locale>
    </locale-config>
  </application>
  <navigation-rule>
    <description>
    </description>
    <from-view-id>/index.xhtml</from-view-id>
    <navigation-case>
      <description>
      </description>
      <from-outcome>success2</from-outcome>
      <to-view-id>/response.jsp</to-view-id>
    </navigation-case>
  </navigation-rule>
  <navigation-rule>
    <description>
    </description>
    <from-view-id>/response.jsp</from-view-id>
    <navigation-case>
      <description>
      </description>
      <from-outcome>success</from-outcome>
      <to-view-id>/index.xhtml</to-view-id>
    </navigation-case>
  </navigation-rule>
  <managed-bean>
    <description>
    </description>
    <managed-bean-name>UserNumberBean</managed-bean-name>
    <managed-bean-class>co.com.guessNumber.UserNumberBean</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
    <managed-property>
      <property-name>minimum</property-name>
      <property-class>double</property-class>
      <value>0</value>
    </managed-property>
    <managed-property>
      <property-name>maximum</property-name>
      <property-class>double</property-class>
      <value>10</value>
    </managed-property>
  </managed-bean>
</faces-config>
```



Modificamos el index.xhtml

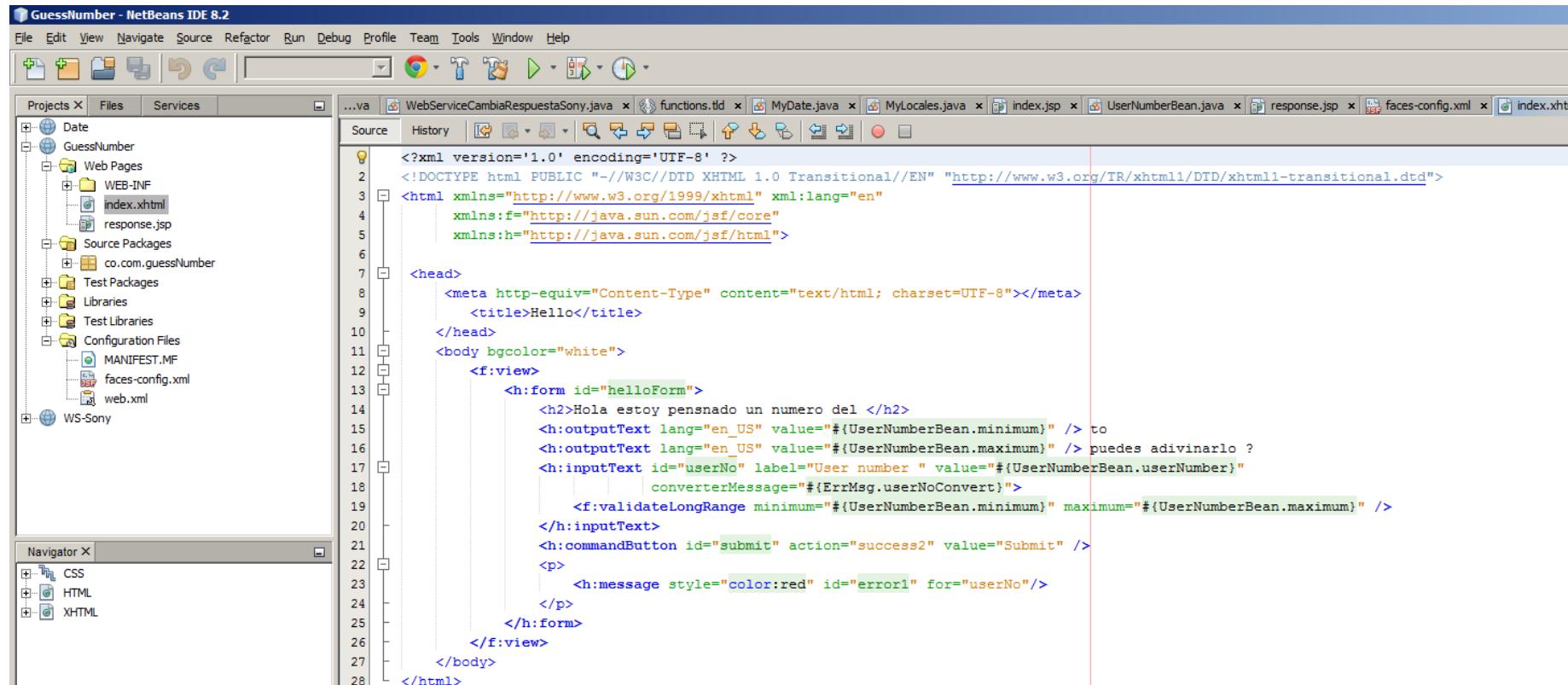


```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html">

    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
        <title>Hello</title>
    </head>
    <body bgcolor="white">
        <f:view>
            <h:form id="helloForm">
                <h2>Hola estoy pensando un numero del </h2>
                <h:outputText lang="en_US" value="#{UserNumberBean.minimum}" /> to
                <h:outputText lang="en_US" value="#{UserNumberBean.maximum}" /> puedes adivinarlo ?
                <h:inputText id="userNo" label="User number" value="#{UserNumberBean.userNumber}"
                             converterMessage="#{ErrMsg.userNoConvert}">
                    <f:validateLongRange minimum="#{UserNumberBean.minimum}" maximum="#{UserNumberBean.maximum}" />
                </h:inputText>
                <h:commandButton id="submit" action="success2" value="Submit" />
                <p>
                    <h:message style="color:red" id="error1" for="userNo"/>
                </p>
            </h:form>
        </f:view>
    </body>
</html>
```



Modificamos el index.xhtml

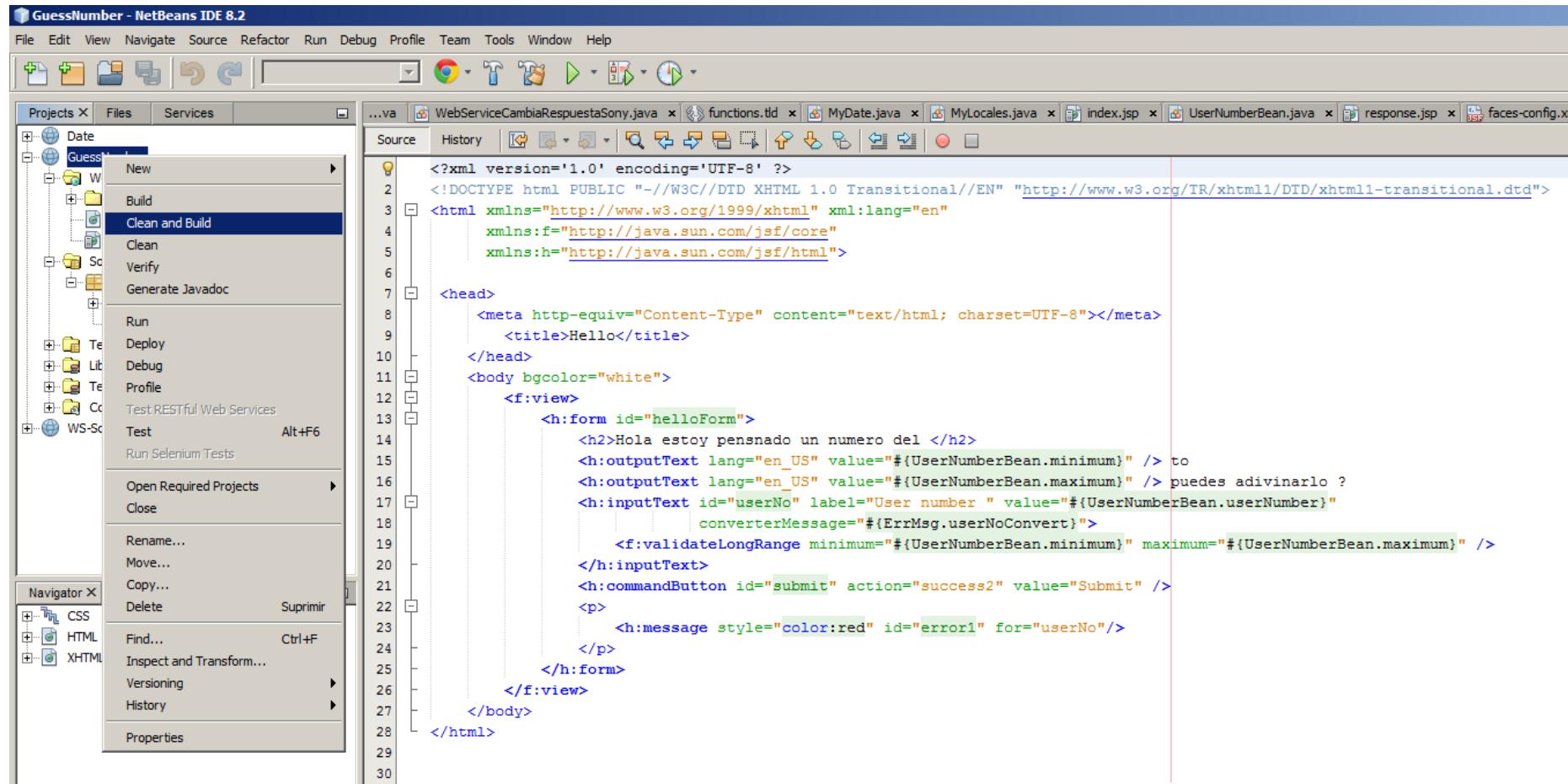


```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html">

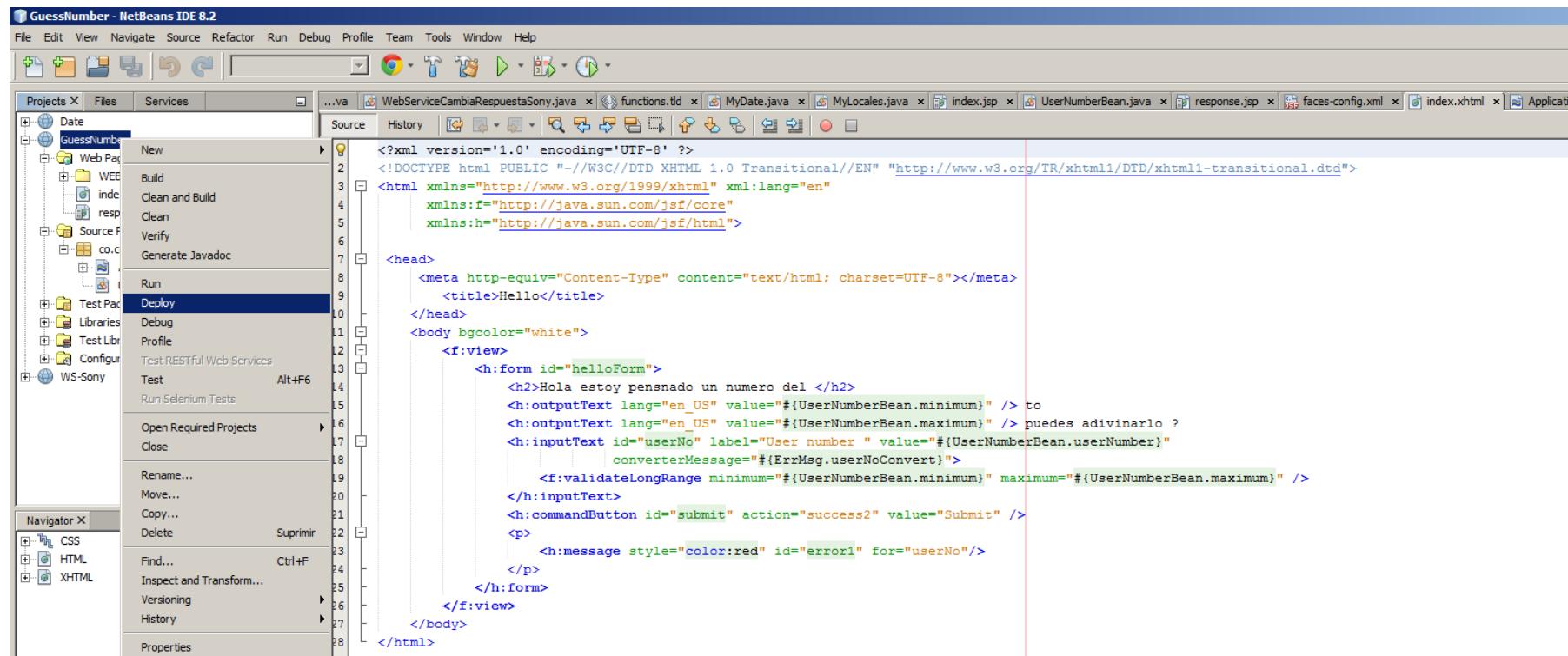
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"></meta>
        <title>Hello</title>
    </head>
    <body bgcolor="white">
        <f:view>
            <h:form id="helloForm">
                <h2>Hola estoy pensando un numero del </h2>
                <h:outputText lang="en_US" value="#{UserNumberBean.minimum}" /> to
                <h:outputText lang="en_US" value="#{UserNumberBean.maximum}" /> puedes adivinarlo ?
                <h:inputText id="userNo" label="User number" value="#{UserNumberBean.userNumber}"
                             converterMessage="#{ErrMsg.userNoConvert}">
                    <f:validateLongRange minimum="#{UserNumberBean.minimum}" maximum="#{UserNumberBean.maximum}" />
                </h:inputText>
                <h:commandButton id="submit" action="success2" value="Submit" />
                <p>
                    <h:message style="color:red" id="error1" for="userNo"/>
                </p>
            </h:form>
        </f:view>
    </body>
</html>
```



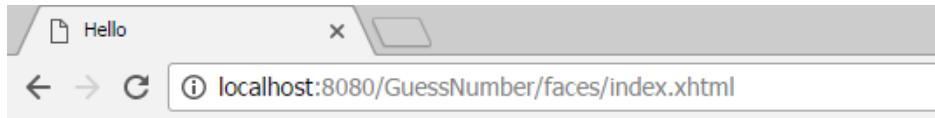
Compilamos el proyecto



Desplegamos el proyecto

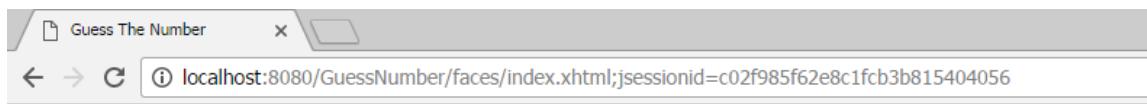


Probamos el proyecto



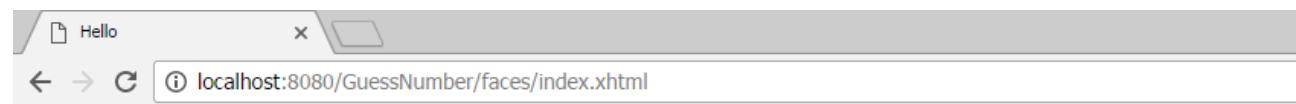
Hola estoy pensando un numero del

0.0 to 10.0 puedes adivinarlo ? Submit



Lo siento, 2 es incorrecto.

[Back](#)



Hola estoy pensando un numero del

0.0 to 10.0 puedes adivinarlo ? Submit

User number : Error de validación: el atributo especificado no está entre los valores esperados: 0 y 10.



Pontificia Universidad
JAVERIANA
Bogotá

Referencias

<http://soapatterns.org>

<https://docs.oracle.com/javaee/7/tutorial/jsf-facelets003.htm>