Servicios Web Seguros

Sesión 4: Optimización de servicios web

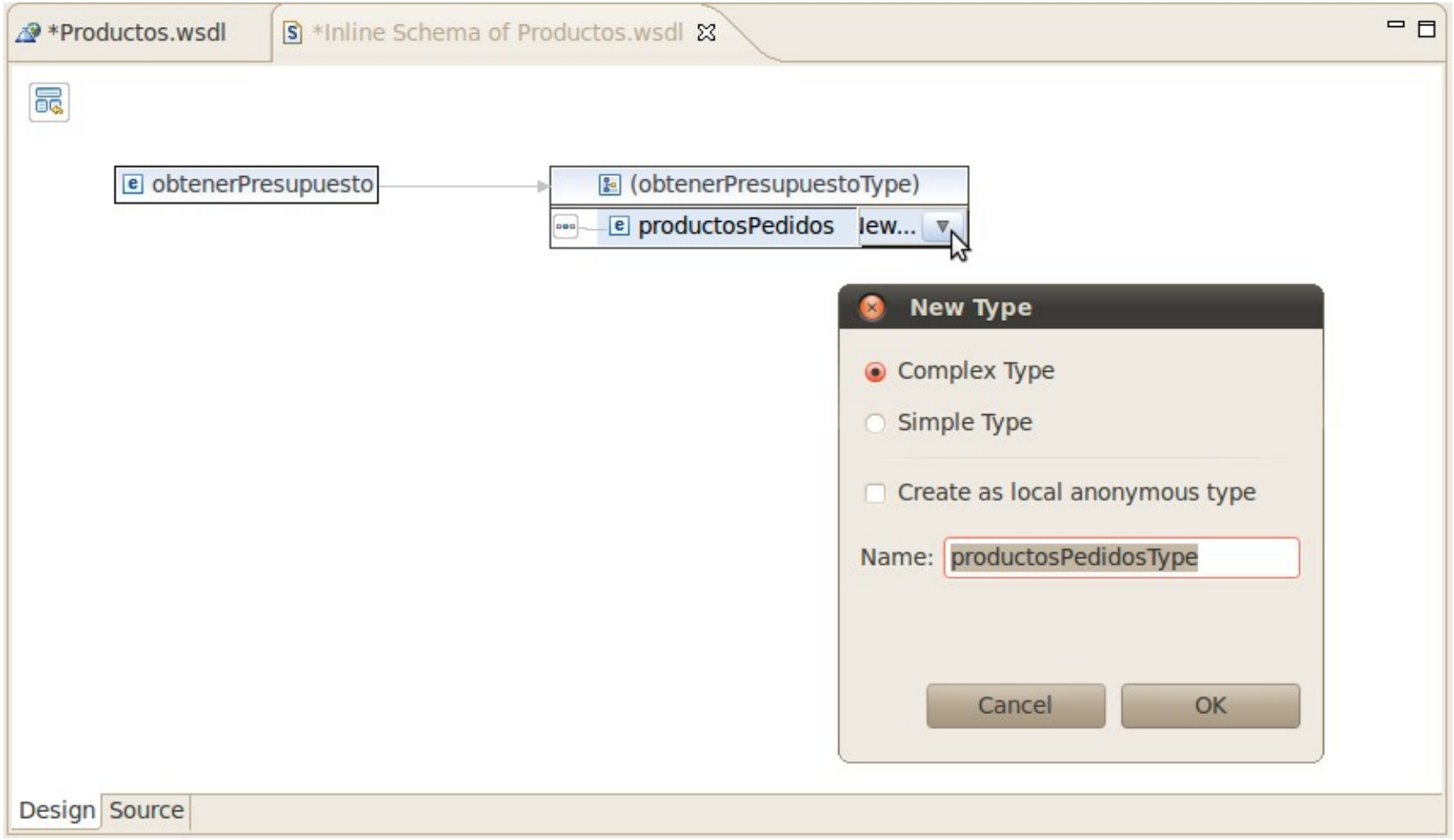
Puntos a tratar

- Estructuras de datos complejas
- Excepciones SOAP
- MTOM

Estructuras de datos complejas

- WSDL
  - Tipos para crear estructuras comlejas
    - <xsd:complexType>
    - <xsd:sequence>
    - <xsd:element>

Estructuras complejas



Multiplicidad

```
<xsd:element name="obtenerPresupuestoResponse">
  <xsd:complexType>
    <xsd:sequence>
          <xsd:element name="productosRespuesta"
                       type="tns:productosRespuestaType"
                       maxOccurs="unbounded" minOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Secuencia de estructuras

- Implementación como List:

```
List<ProductosRespuestaType> prs =
  new ArrayList<ProductosRespuestaType>();
for(ProductosPedidosType prPedido : productosPedidos){
        ProductosRespuestaType pr = new ProductosRespuestaType();
        pr.setIdProducto(prPedido.getIdProducto());
        pr.setImporte( ... );
        pr.setFoto( ... );
        prs.add(pr);
}
return prs;
```

Envío de excepciones SOAP

- Definición en el WSDL

```
    <xs:element name="ObtenerPresupuestoFault" type="tns:fault"/>
    <xs:complexType name="ObtenerPresupuestoFault">
      <xs:sequence>
        <xs:element name="errorMessage" type="xs:string" minOccurs="1"
maxOccurs="1" nillable="false"/>
        <xs:element name="errorCode" type="xs:int" minOccurs="1"
maxOccurs="1" nillable="false"/>
      </xs:sequence>
    </xs:complexType>
```

Envío de excepciones SOAP

- Lanzar la excepción desde el servicio
- throw new ObtenerPresupuestoFault_Exception("No existe el producto "+prPedido.getIdProducto( ));

```
Invoking preguntaPrecios...
Exception in thread "main" javax.xml.ws.soap.SOAPFaultException:
                                    No existe el producto Domo Kun
        at org.apache.cxf.jaxws.JaxWsClientProxy.invoke(
                                              JaxWsClientProxy.java:146)
        at $Proxy40.obtenerPresupuesto(Unknown Source)
        at es.ua.jtech.productos.custom.Productos_ProductosSOAP_Client.
                          main(Productos_ProductosSOAP_Client.java:80)
Caused by: org.apache.cxf.binding.soap.SoapFault:
                                    No existe el producto Domo Kun
[...]
```

Envío de excepciones SOAP

- Cuando el cliente recibe una excepción, ésta debe ser capturada.
- La excepción impide la recepción de otros datos que el servidor haya estado introduciendo en el mensaje.

Envío de archivos / datos

- Tipos binarios en el WSDL
  - <xsd:element name="foto" type="xsd:base64Binary" />

Envío de archivos / datos

- Cargar datos

```
ByteArrayOutputStream baos =
  new ByteArrayOutputStream();
byte[] buffer = new byte[1024];
try {
        int nbytes;
        while(true){
                if((nbytes=fis.read(buffer)) == -1){
                        break;
                }
                        baos.write(buffer,0,nbytes);
        }
        pr.setFoto(baos.toByteArray());
} catch (IOException e) {
        e.printStackTrace();
        pr.setFoto(null);
}
```

Envío de archivos / datos

- Obtener los datos

```
if(pr.getFoto()!=null){
        try{
                String filename = pr.getIdProducto()+".jpg";
                FileOutputStream fos =
                  new FileOutputStream(filename);
                fos.write(pr.getFoto());
                fos.close();
                //Ejecutar el visor Eye of Gnome
                Runtime.getRuntime().exec("eog -n "+filename);
        }catch(IOException e){
                System.out.println("Error escribiendo la imagen: "
                  + pr.getIdProducto());
        }
}
```

Optimización MTOM

- W3C Message Transmission Optimization Mechanism
  - Envío eficiente de datos binarios
  - Soporte tipos MIME
  - Se usa con XOP: XML-binary Optimized Packaging
- En el WSDL:
  - Namespace: xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
  - <xsd:element name="foto" type="xsd:base64Binary" xmime:expectedContentTypes="application/octet-stream"/>

Optimización MTOM

- Envío de los datos:
  - DataHandler dh = null;
  - dh = new DataHandler(new FileDataSource("src/main/resources/fotos/calc.jpg"));
  - pr.setFoto(dh);
- DataHandler gestiona de forma eficiente los flujos de entrada sin cargarlos enteros en memoria (no se carga todo en un buffer).

Optimización MTOM

- Obtener los datos
  - FileOutputStream fos = new FileOutputStream(filename);
  - pr.getFoto().writeTo(fos);

¿Preguntas...?