

CURSO JAVA EE

SGA CON REST WEBSERVICES (JAX-RS)



Por el experto: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

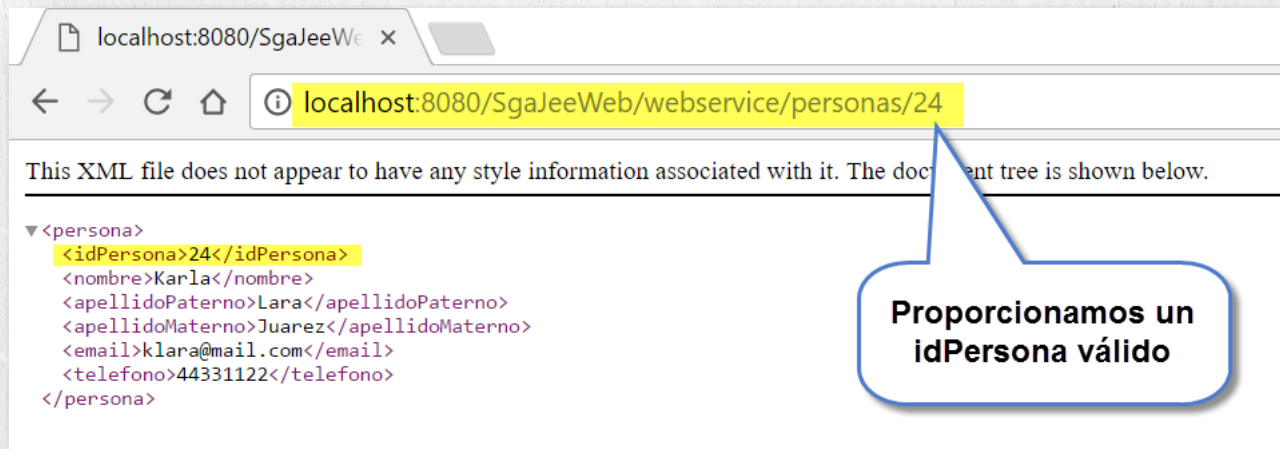


CURSO JAVA EE

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

El objetivo del ejercicio exponer el método listarPersonas, registrarPersona, modificarPersona, eliminarPersona del EJB del proyecto SGA utilizando Rest Web Services con ayuda del API JAX-RS. El resultado se muestra a continuación:

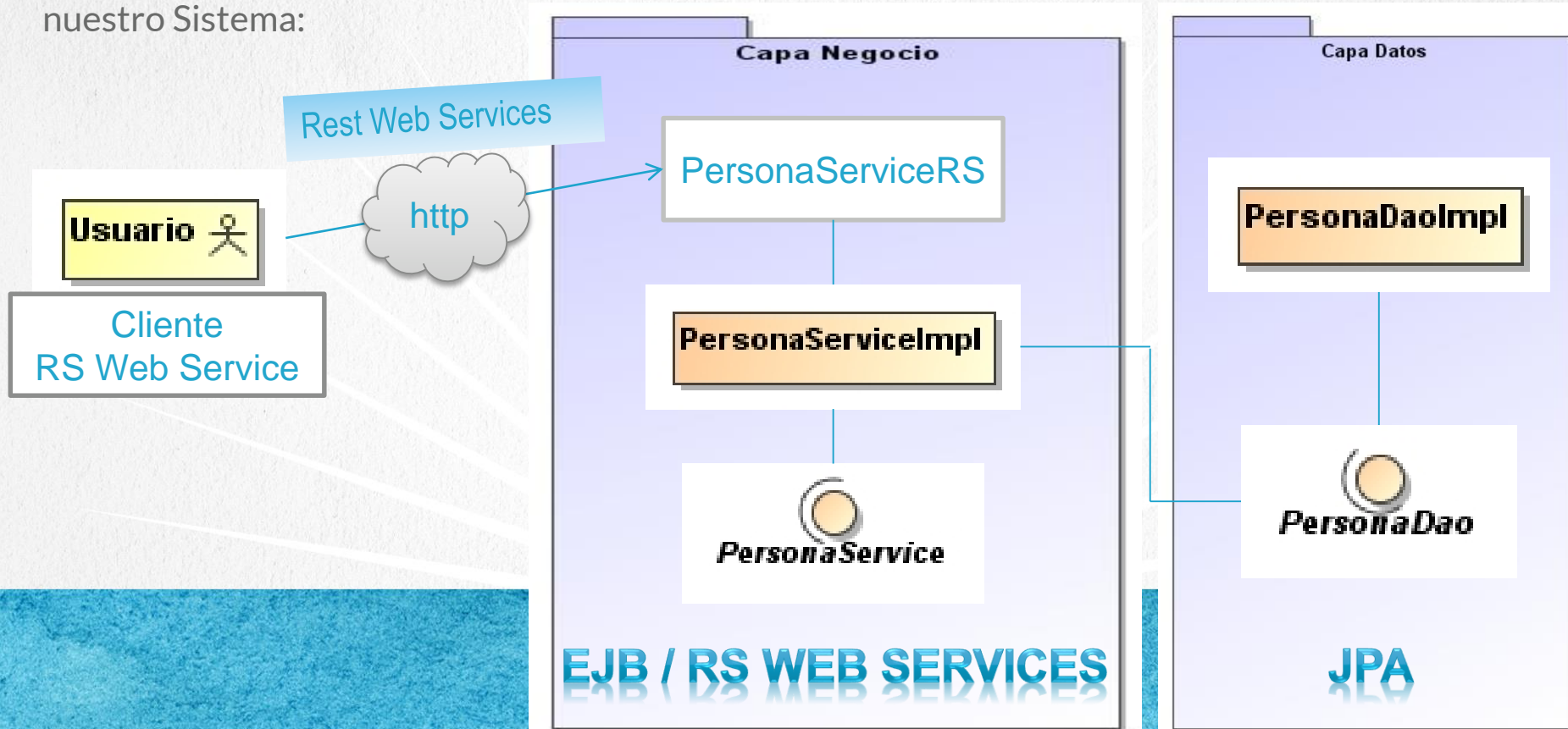


CURSO JAVA EE

www.globalmentoring.com.mx

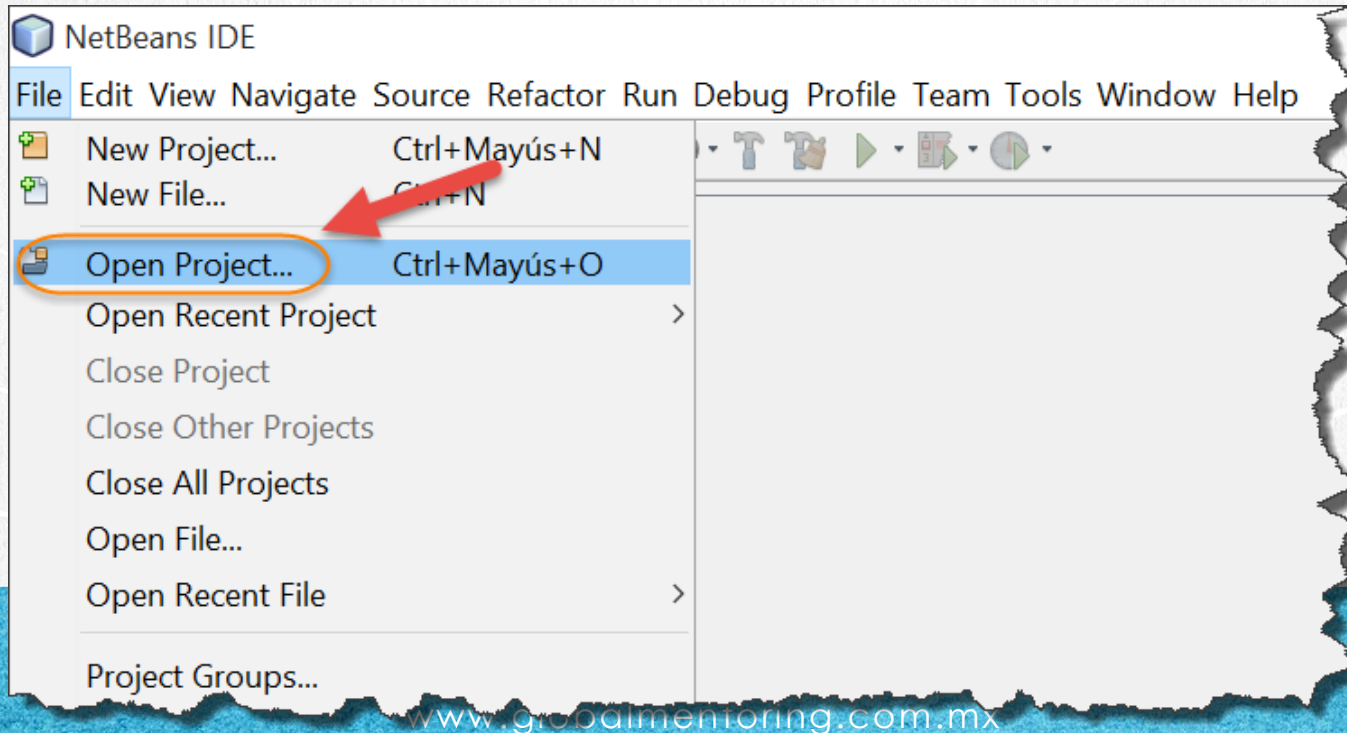
ARQUITECTURA SGA CON WEB SERVICES

Este es el Diagrama de Clases del Ejercicio, donde se pueden observar la Arquitectura de nuestro Sistema:



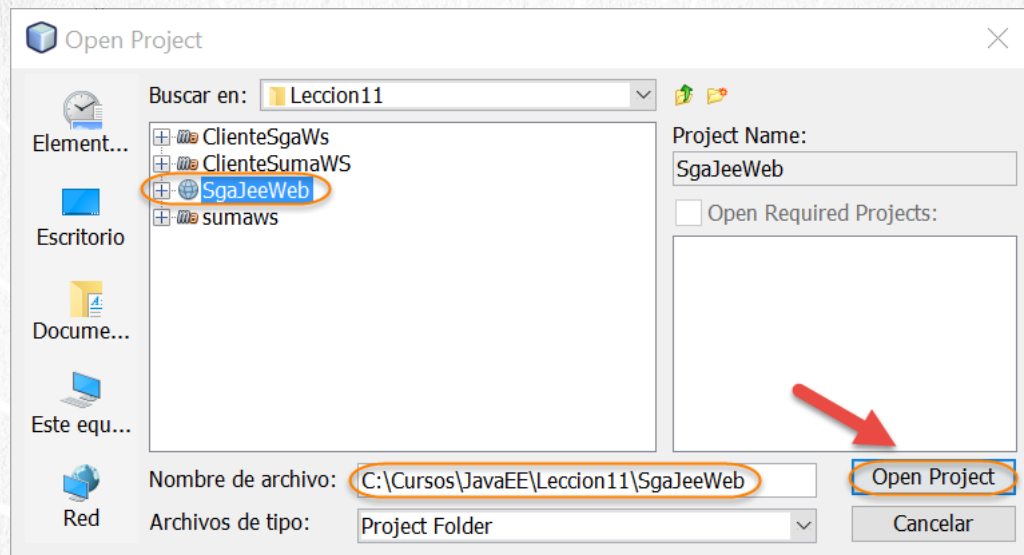
PASO 1. ABRIMOS EL PROYECTO

En caso que no tengamos abierto el proyecto SgaJeeWeb lo abrimos:



PASO 1. ABRIMOS EL PROYECTO

En caso que no tengamos abierto el proyecto SgaJeeWeb lo abrimos:

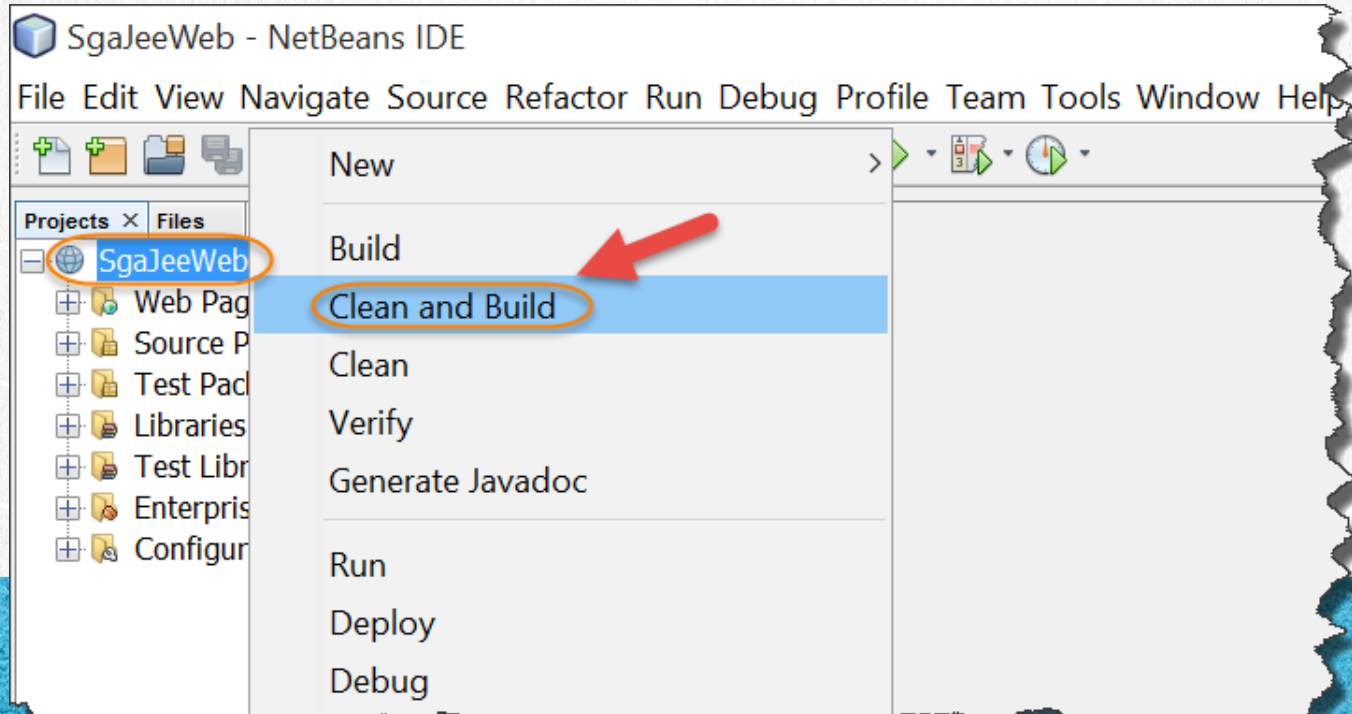


CURSO JAVA EE

www.globalmentoring.com.mx

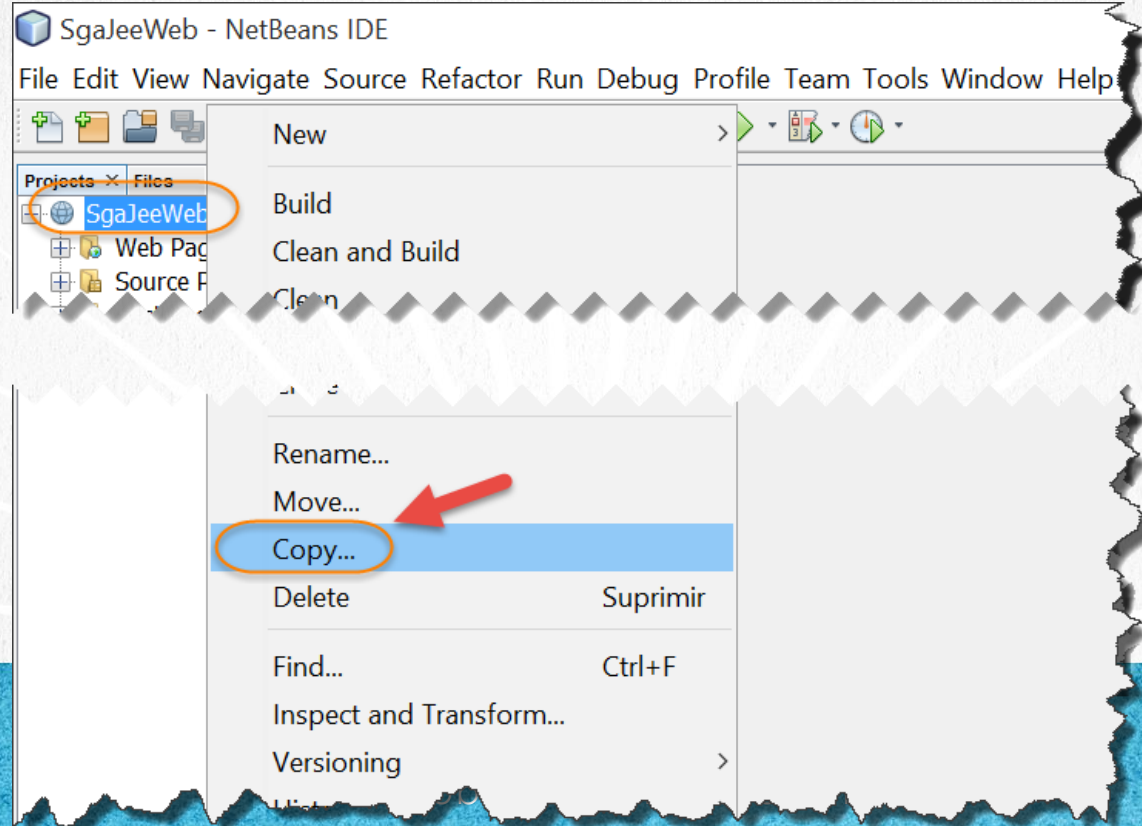
PASO 1. ABRIMOS EL PROYECTO

Esperamos a que cargue completamente el proyecto. En caso que marque error el proyecto, hacemos un Clean & Build para que se muestren todos los archivos, este paso es opcional:



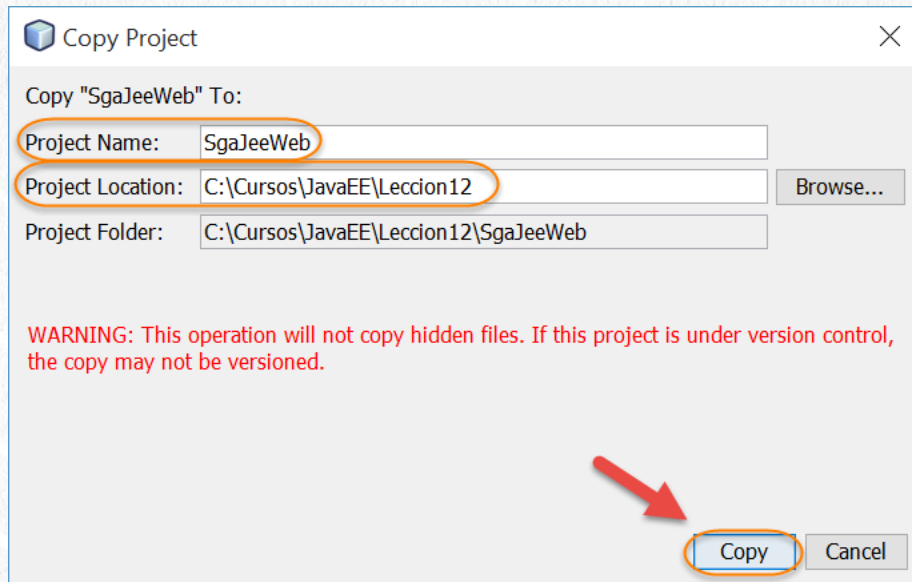
PASO 2. COPIAMOS EL PROYECTO

Copiamos el proyecto para ponerlo en la nueva ruta:



PASO 2. COPIAMOS EL PROYECTO

Copiamos el proyecto para ponerlo en la nueva ruta:

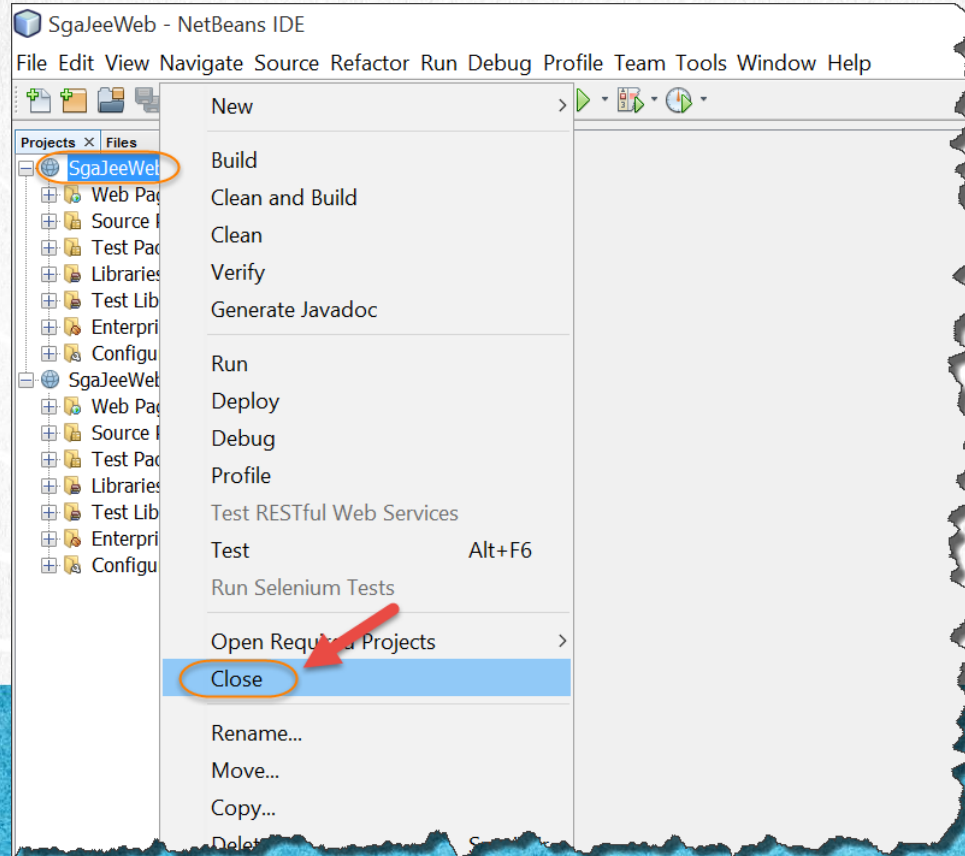


CURSO JAVA EE

www.globalmentoring.com.mx

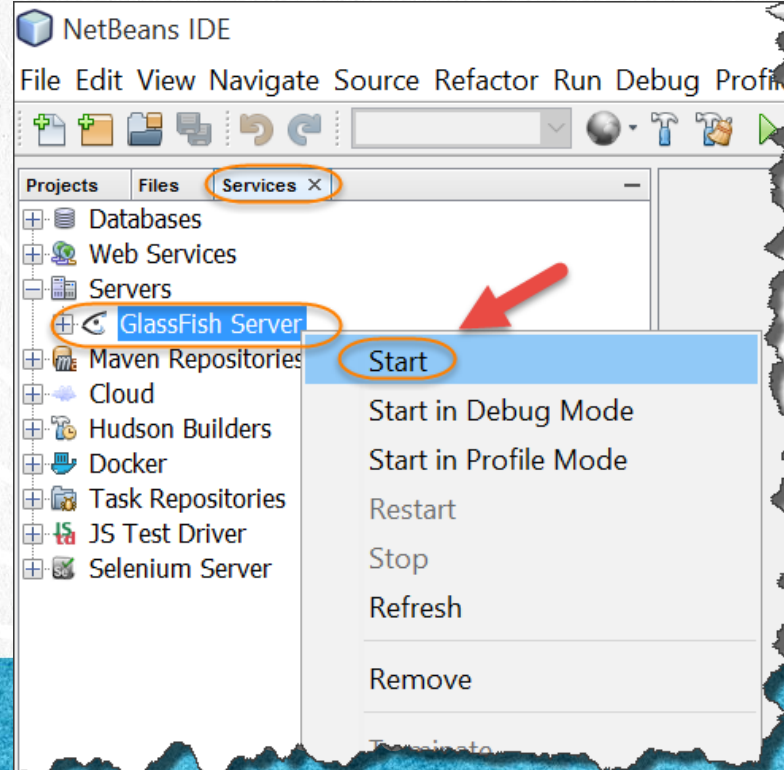
PASO 2. CERRAMOS EL PROYECTO

Cerramos el proyecto anterior y dejamos solo el nuevo:



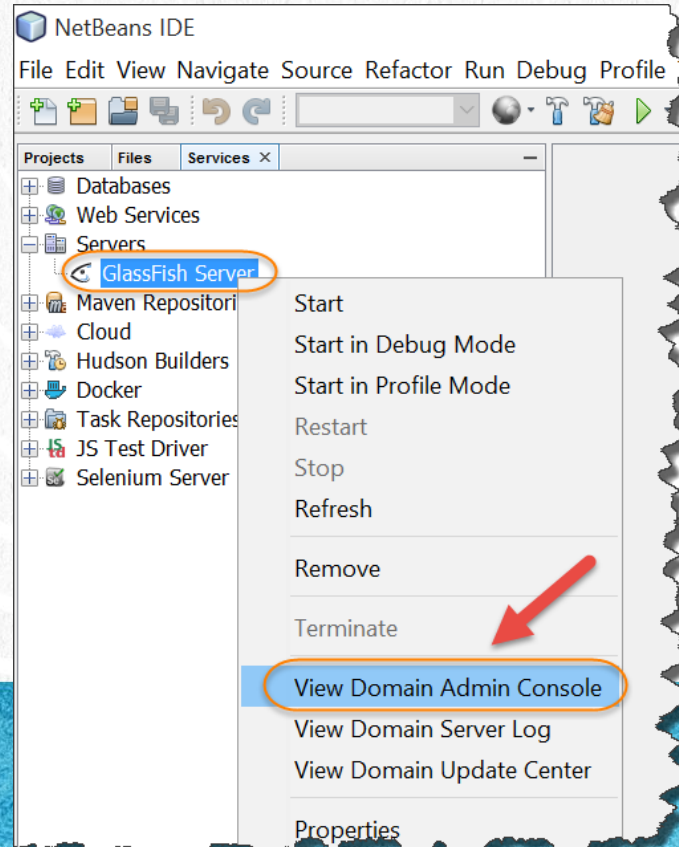
PASO 3. HACEMOS UNDEPLOY EN GLASSFISH

Hacemos undeploy de los proyectos que tengamos en Glassfish:



PASO 3. HACEMOS UNDEPLOY EN GLASSFISH

Hacemos undeploy de los proyectos que tengamos en Glassfish:



PASO 3. HACEMOS UNDEPLOY EN GLASSFISH

Hacemos undeploy de los proyectos que tengamos en Glassfish. Esto lo hacemos para que los EJB que estén desplegados no choquen con los que vamos a subir y también el log de Glassfish esté lo más limpio posible:

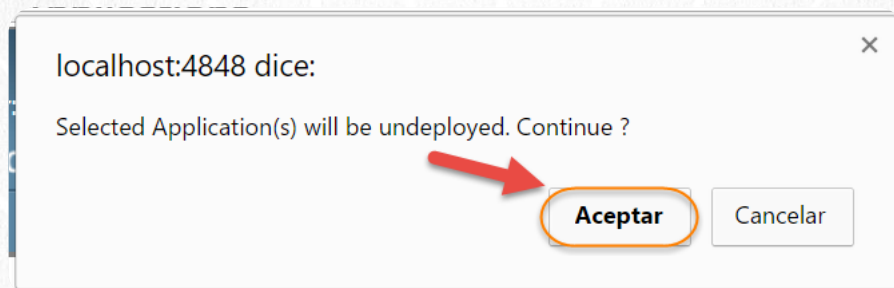
The screenshot shows the GlassFish Administration Console interface. The browser address bar indicates the URL is `localhost:4848/common/index.jsf`. The page title is "GlassFish™ Server Open Source Edition". The left sidebar, titled "Tree", shows a navigation menu with "Applications" selected, marked with a red circle and the number 1. The main content area, titled "Applications", contains a section for "Deployed Applications (1)". This section includes a toolbar with buttons for "Deploy...", "Undeploy" (circled in red with a red circle and the number 3), "Enable", and "Disable". Below the toolbar is a table with the following data:

Select	Name	Deployment Order	Enabled	Engines	Action
<input checked="" type="checkbox"/>	SgaJeeWeb	100	✓	ejb, webservices, web	Launch Redeploy Reload

The "SgaJeeWeb" application is highlighted with a red circle and the number 2. The "Undeploy" button is circled in red with a red circle and the number 3. A red arrow points from the "Undeploy" button to the "SgaJeeWeb" application row.

PASO 3. HACEMOS UNDEPLOY EN GLASSFISH

Hacemos undeploy de los proyectos que tengamos en Glassfish.

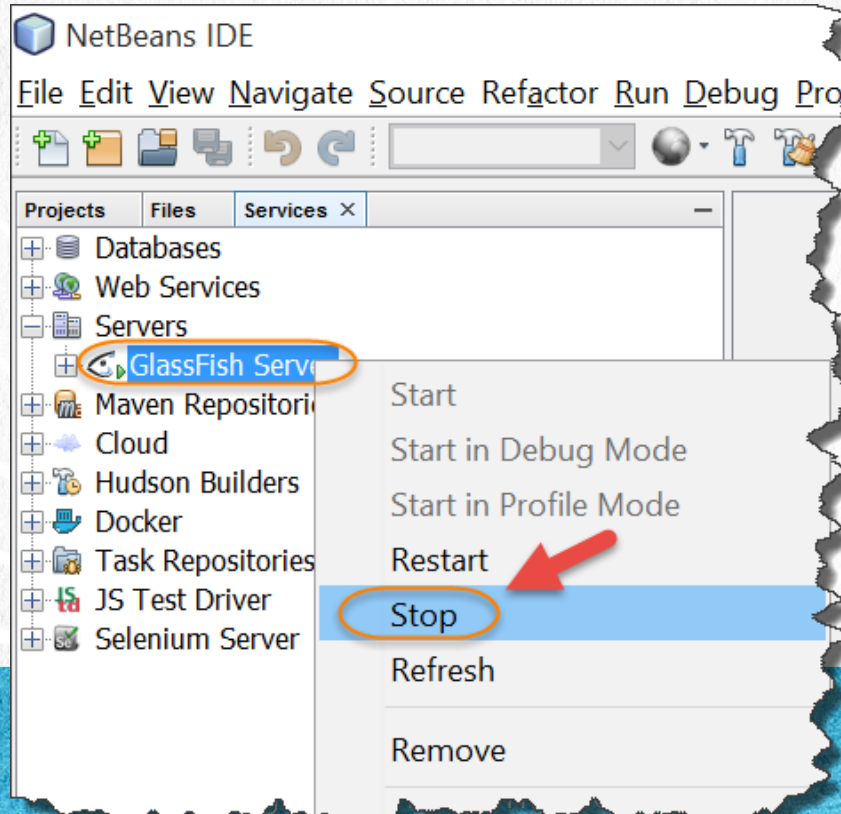


CURSO JAVA EE

www.globalmentoring.com.mx

PASO 4. DETENEMOS GLASSFISH

Detenemos el servidor de Glassfish:

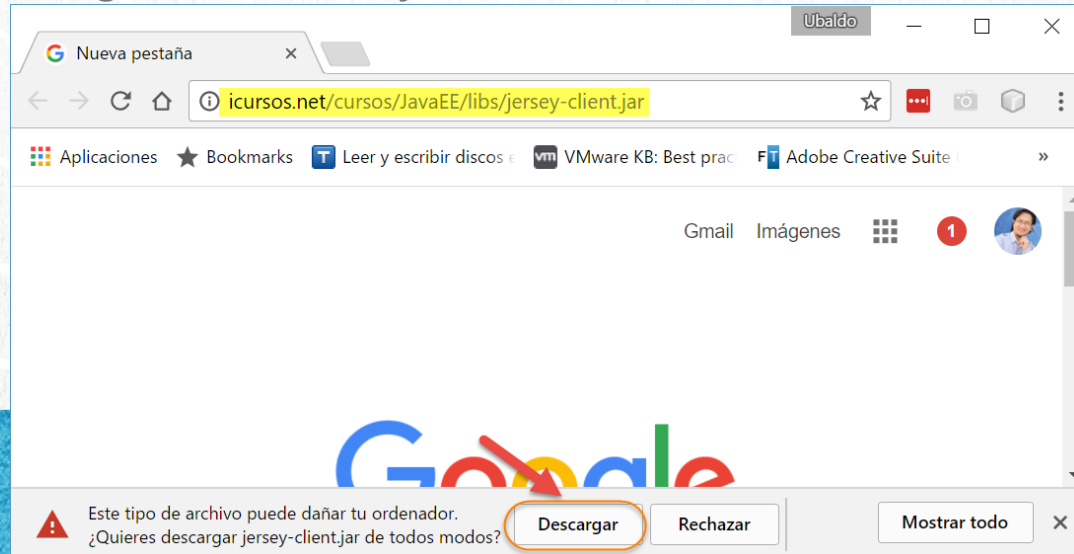


PASO 5. AGREGAMOS UNA LIBRERÍA .JAR

Agregamos la librería jersey-client.jar. La descargamos del siguiente link y la agregamos a las librerías de nuestro proyecto:

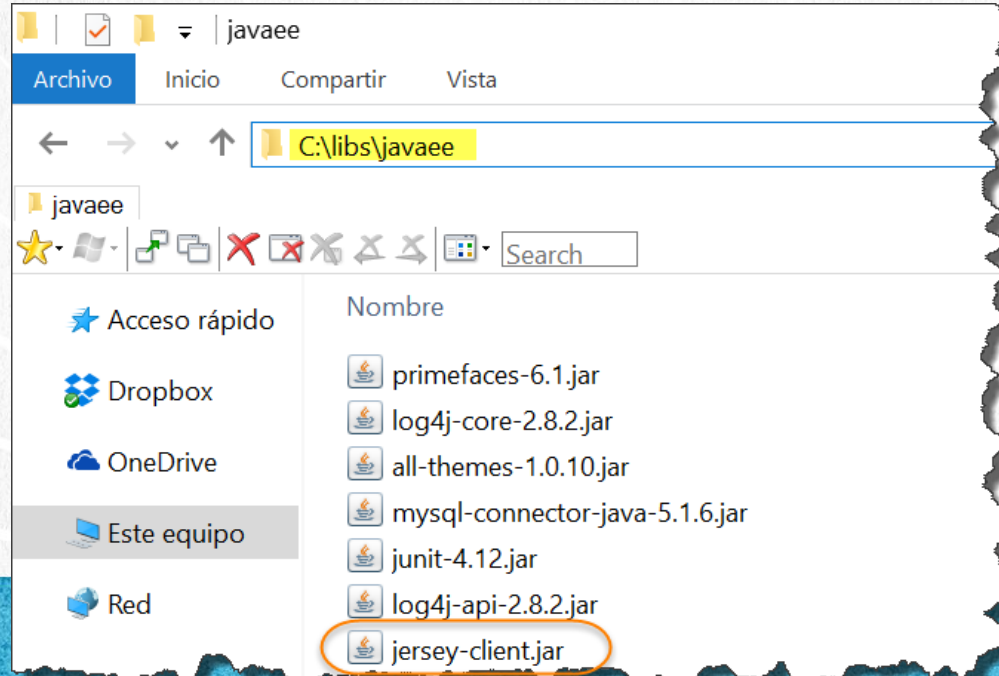
<http://icursos.net/cursos/JavaEE/libs/jersey-client.jar>

Aceptamos la descarga del archivo .jar



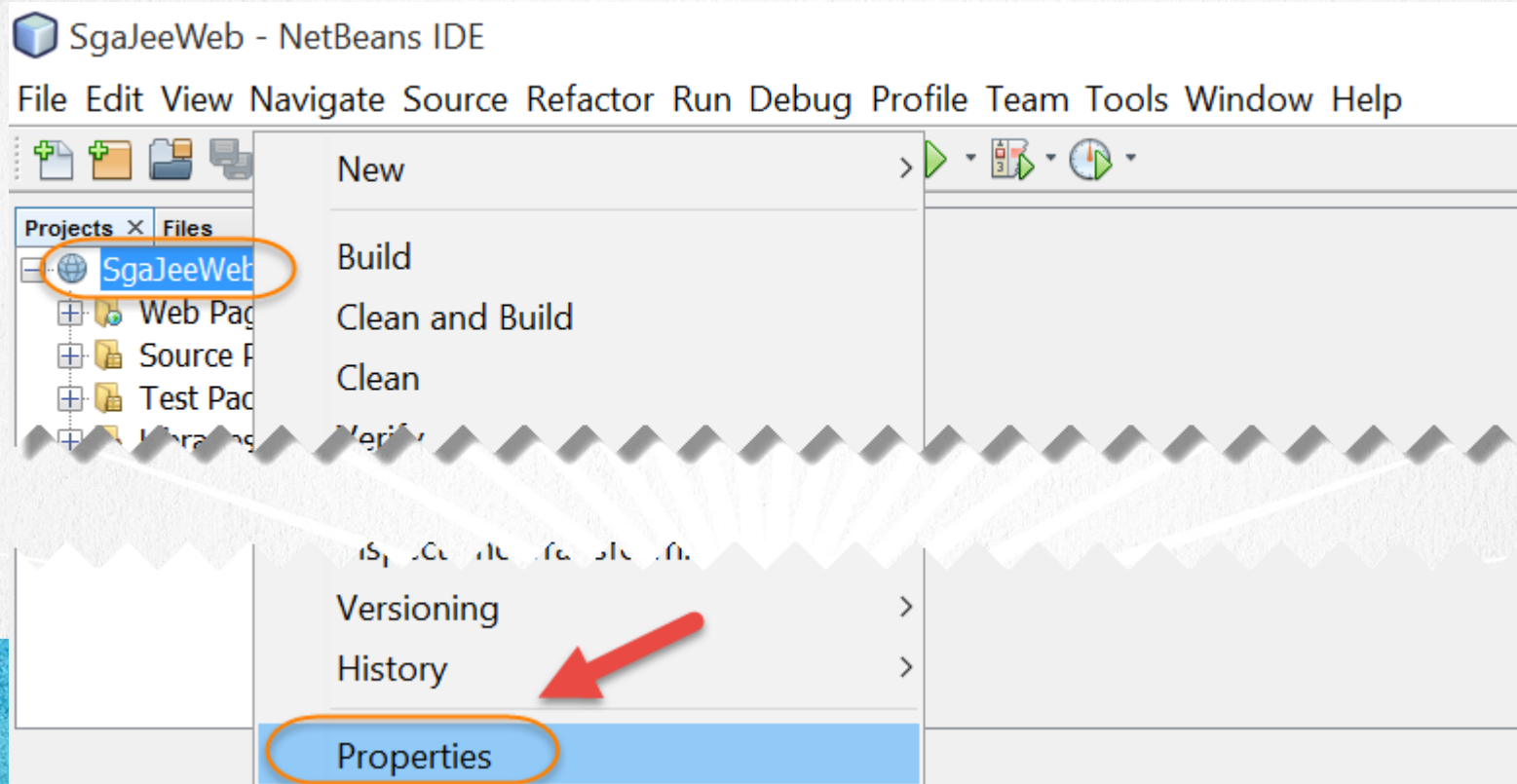
PASO 5. AGREGAMOS UNA LIBRERÍA .JAR

Agregamos el archivo a las librerías del proyecto. Las versiones utilizadas pueden ser más recientes a las mostradas. Movemos el .jar descargado a la ruta: C:\libs\javaee:



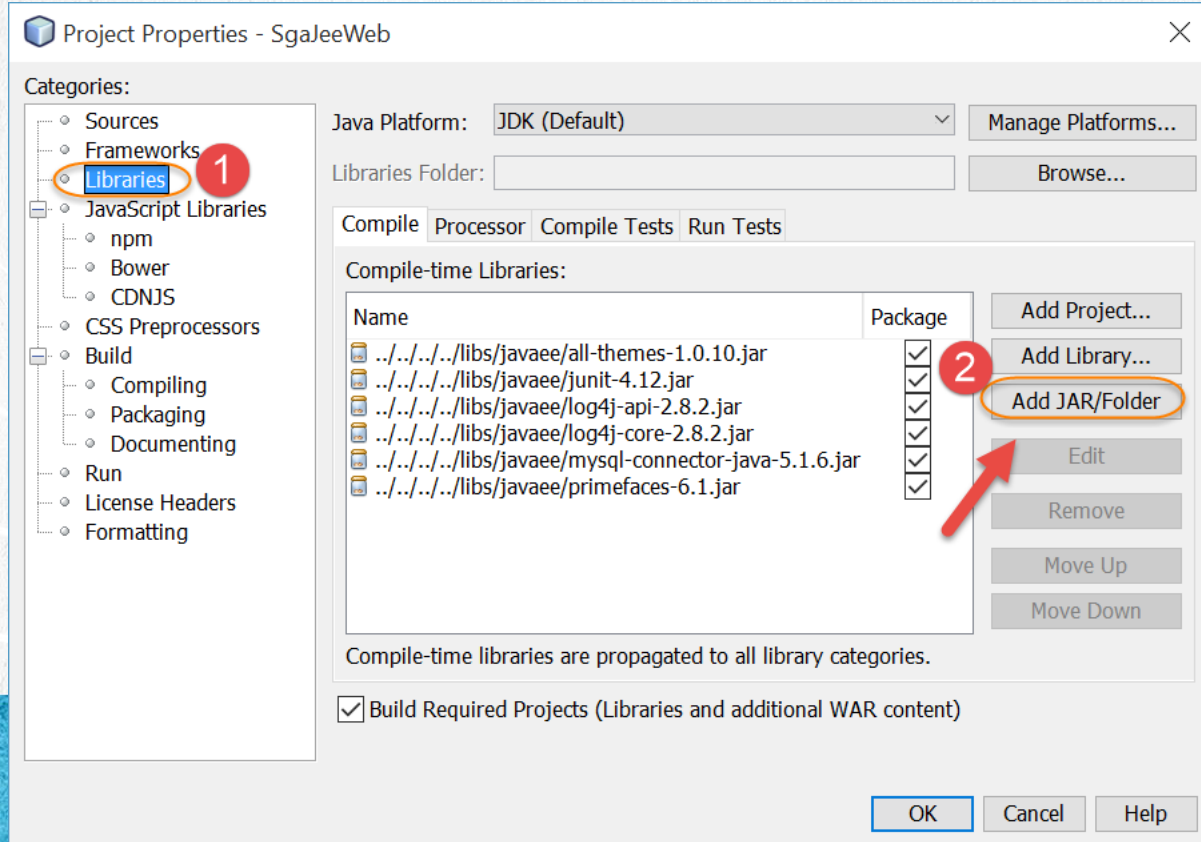
PASO 5. AGREGAMOS UNA LIBRERÍA .JAR

Agregamos la librería al proyecto:



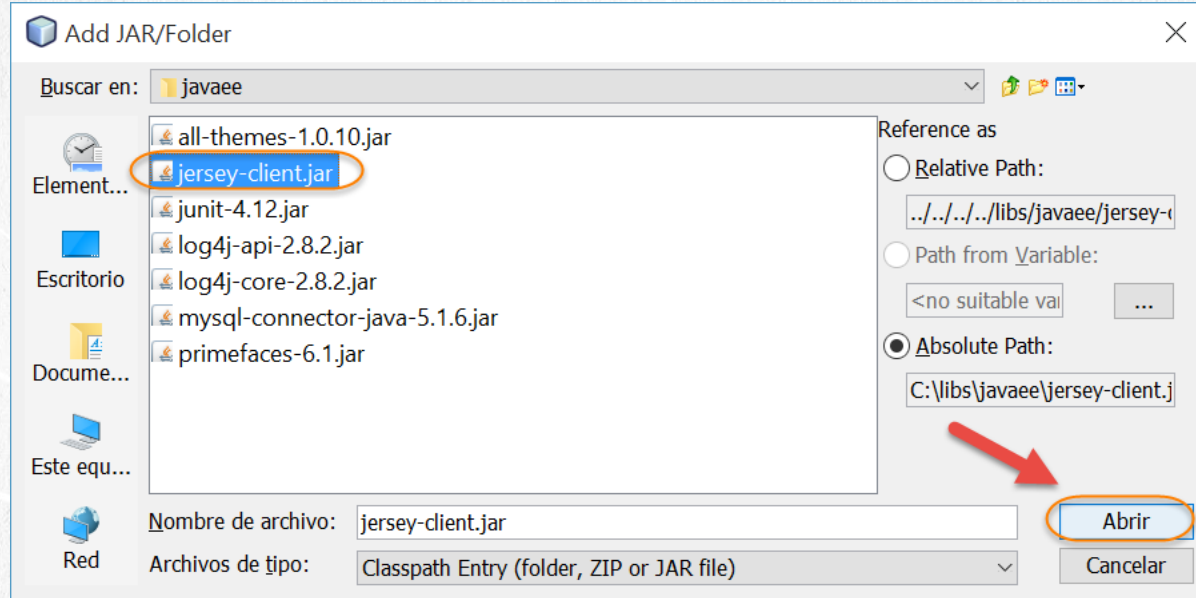
PASO 5. AGREGAMOS UNA LIBRERÍA .JAR

Agregamos la librería al proyecto:



PASO 5. AGREGAMOS UNA LIBRERÍA .JAR

Agregamos la librería al proyecto:

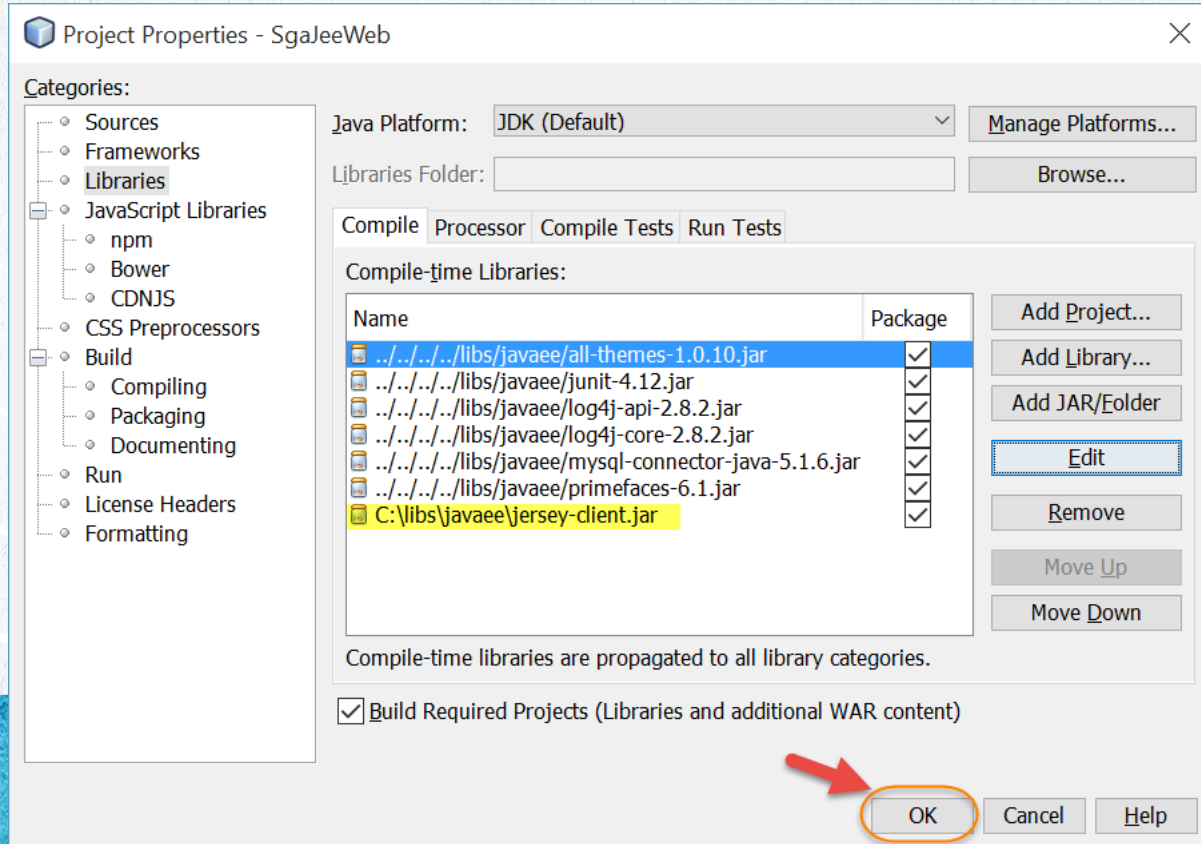


CURSO JAVA EE

www.globalmentoring.com.mx

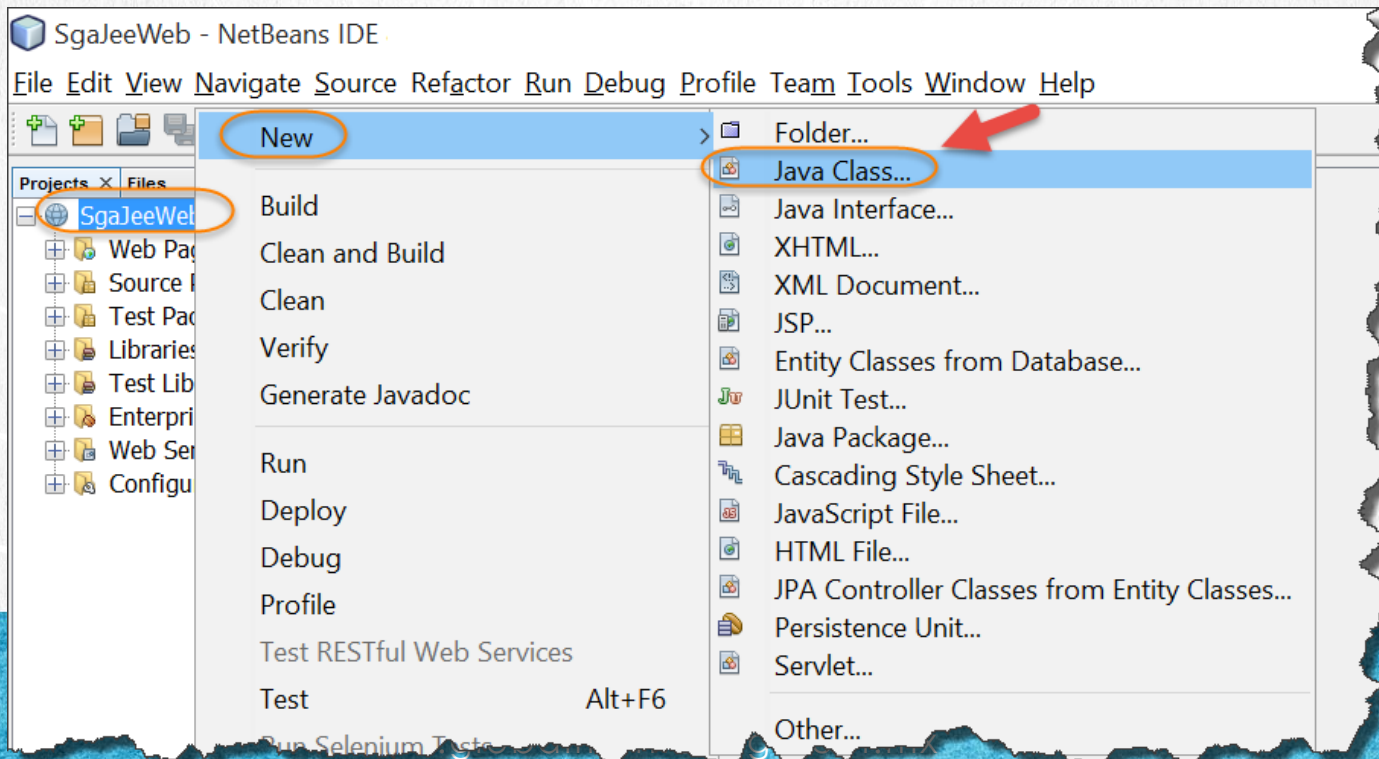
PASO 5. AGREGAMOS UNA LIBRERÍA .JAR

Agregamos la librería al proyecto:



PASO 6. CREAMOS UN ARCHIVO JAVA

Creamos la clase PersonaServiceRS.java para exponer los métodos los métodos de listar, agregar, modificar y eliminar Personas vía Rest Web Services :



PASO 6. CREAMOS UN ARCHIVO JAVA

Creamos la clase PersonaServiceRS.java:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: PersonaServiceRS

Project: SgaJeeWeb

Location: Source Packages

Package: mx.com.gm.sga.servicio.rest

Created File: ccion12\SgaJeeWeb\src\java\mx\com\gm\sga\servicio\rest\PersonaServiceRS.java

< Back Next > **Finish** Cancel Help

CURSO JAVA EE

www.globalmentoring.com.mx

PASO 7. MODIFICAMOS EL CÓDIGO

[Archivo PersonaServiceRS.java:](#)

Dar click para ir al código

```
package mx.com.gm.sga.servicio.rest;

import java.util.List;

import javax.ejb.EJB;
import javax.ejb.Stateless;
import javax.ws.rs.Consumes;
import javax.ws.rs.DELETE;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.Response.Status;
import mx.com.gm.sga.domain.Persona;
import mx.com.gm.sga.servicio.PersonaService;

@Path("/personas")
@Stateless
public class PersonaServiceRS {

    @EJB
    private PersonaService personaService;
```

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo PersonaServiceRS.java:

Dar click para ir al código

```
@GET
@Produces(value={MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
public List<Persona> listarPersonas() {
    return personaService.listarPersonas();
}

@GET
@Produces(value={MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
@Path("/{id}") //hace referencia a /personas/{id}
public Persona encontrarPersonaPorId(@PathParam("id") int id) {
    return personaService.encontrarPersonaPorId(new Persona(id));
}

@POST
@Produces(value={MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
@Consumes(value={MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
public Response agregarPersona(Persona persona) {
    try {
        personaService.registrarPersona(persona);
        return Response.ok().entity(persona).build();
    } catch (Exception e) {
        return Response.status(Status.INTERNAL_SERVER_ERROR).build();
    }
}
```

CURSO DE JAVA EE

www.globalmentoring.com.mx

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo PersonaServiceRS.java:

Dar click para ir al código

```
@PUT
@Produces(value={MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
@Consumes(value={MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
@Path("/{id}")
public Response modificarPersona(@PathParam("id") int id, Persona personaModificada) {
    try {
        Persona persona = personaService.encontrarPersonaPorId(new Persona(id));
        if (persona != null) {
            personaService.modificarPersona(personaModificada);
            return Response.ok().entity(personaModificada).build();
        } else {
            return Response.status(Status.NOT_FOUND).build();
        }
    } catch (Exception e) {
        return Response.status(Status.INTERNAL_SERVER_ERROR).build();
    }
}

@DELETE
@Path("/{id}")
public Response eliminarPersonaPorId(@PathParam("id") int id) {
    try {
        personaService.eliminarPersona(new Persona(id));
        return Response.ok().build();
    } catch (Exception e) {
        return Response.status(404).build();
    }
}
```


PASO 8. MODIFICAMOS UN ARCHIVO JAVA

- Modificamos la clase de dominio Persona, agregando la siguiente anotación al inicio de la clase:

@XmlElement

Quedando la clase como sigue:

```
@Entity
@Table(name = "persona")
@NamedQueries({
    @NamedQuery(name = "Persona.findAll", query = "SELECT p FROM Persona p")
    , @NamedQuery(name = "Persona.findByIdPersona", query = "SELECT p FROM Persona p WHERE p.idPersona = :idPersona")
    , @NamedQuery(name = "Persona.findByName", query = "SELECT p FROM Persona p WHERE p.nombre = :nombre")
    , @NamedQuery(name = "Persona.findByApellidoPaterno", query = "SELECT p FROM Persona p WHERE p.apellidoPaterno = :apellidoPaterno")
    , @NamedQuery(name = "Persona.findByApellidoMaterno", query = "SELECT p FROM Persona p WHERE p.apellidoMaterno = :apellidoMaterno")
    , @NamedQuery(name = "Persona.findByEmail", query = "SELECT p FROM Persona p WHERE p.email = :email")
    , @NamedQuery(name = "Persona.findByTelefono", query = "SELECT p FROM Persona p WHERE p.telefono = :telefono")})
@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class Persona implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "id_persona")
    private Integer idPersona;
```

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

Dar click para ir al código

```
package mx.com.gm.sga.domain;

import java.io.Serializable;
import java.util.List;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlTransient;

@Entity
@Table(name = "persona")
@NamedQueries({
    @NamedQuery(name = "Persona.findAll", query = "SELECT p FROM Persona p")
    , @NamedQuery(name = "Persona.findByIdPersona", query = "SELECT p FROM Persona p WHERE p.idPersona = :idPersona")
    , @NamedQuery(name = "Persona.findByName", query = "SELECT p FROM Persona p WHERE p.nombre = :nombre")
    , @NamedQuery(name = "Persona.findByIdApellidoPaterno", query = "SELECT p FROM Persona p WHERE p.apellidoPaterno = :apellidoPaterno")
    , @NamedQuery(name = "Persona.findByIdApellidoMaterno", query = "SELECT p FROM Persona p WHERE p.apellidoMaterno = :apellidoMaterno")
    , @NamedQuery(name = "Persona.findByEmail", query = "SELECT p FROM Persona p WHERE p.email = :email")
    , @NamedQuery(name = "Persona.findByTelefono", query = "SELECT p FROM Persona p WHERE p.telefono = :telefono")})
```

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

Dar click para ir al código

```
@XmlElement
@XmlAccessorType(XmlAccessType.FIELD)
public class Persona implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "id_persona")
    private Integer idPersona;

    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 45)
    @Column(name = "nombre")
    private String nombre;

    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 45)
    @Column(name = "apellido_paterno")
    private String apellidoPaterno;

    @Size(max = 45)
    @Column(name = "apellido_materno")
    private String apellidoMaterno;

    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 45)
    @Column(name = "email")
    private String email;
```


PASO 8. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

Dar click para ir al código

```
@Size(max = 45)
@Column(name = "telefono")
private String telefono;

@XmlTransient
@OneToMany(mappedBy = "persona", cascade = CascadeType.ALL)
private List<Usuario> usuarios;

public Persona() {
}

public Persona(Integer idPersona) {
    this.idPersona = idPersona;
}

public Persona(String nombre, String apePaterno, String apeMaterno,
    String email, String telefono) {
    this.nombre = nombre;
    this.apellidoPaterno = apePaterno;
    this.apellidoMaterno = apeMaterno;
    this.email = email;
    this.telefono = telefono;
}

public Persona(Integer idPersona, String nombre, String apellidoPaterno, String email) {
    this.idPersona = idPersona;
    this.nombre = nombre;
    this.apellidoPaterno = apellidoPaterno;
    this.email = email;
}
```

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

Dar click para ir al código

```
public Persona(Integer idPersona, String nombre, String apellidoPaterno, String apellidoMaterno,
               String email, String telefono) {
    this.idPersona = idPersona;
    this.nombre = nombre;
    this.apellidoPaterno = apellidoPaterno;
    this.apellidoMaterno = apellidoMaterno;
    this.email = email;
    this.telefono = telefono;
}

public Integer getIdPersona() {
    return idPersona;
}

public void setIdPersona(Integer idPersona) {
    this.idPersona = idPersona;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getApellidoPaterno() {
    return apellidoPaterno;
}

public void setApellidoPaterno(String apellidoPaterno) {
    this.apellidoPaterno = apellidoPaterno;
}
```

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

Dar click para ir al código

```
public String getApellidoMaterno() {  
    return apellidoMaterno;  
}  
  
public void setApellidoMaterno(String apellidoMaterno) {  
    this.apellidoMaterno = apellidoMaterno;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getTelefono() {  
    return telefono;  
}  
  
public void setTelefono(String telefono) {  
    this.telefono = telefono;  
}  
  
public List<Usuario> getUsuarios() {  
    return usuarios;  
}  
  
public void setUsuarios(List<Usuario> usuarios) {  
    this.usuarios = usuarios;  
}
```


PASO 8. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

Dar click para ir al código

```
@Override
public int hashCode() {
    int hash = 0;
    hash += (idPersona != null ? idPersona.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    if (!(object instanceof Persona)) {
        return false;
    }
    Persona other = (Persona) object;
    if ((this.idPersona == null && other.idPersona != null) || (this.idPersona != null && !this.idPersona.equals(other.idPersona))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Persona{" + "idPersona=" + idPersona + ", nombre=" + nombre + ", apellidoPaterno=" + apellidoPaterno + ", apellidoMaterno=" +
    apellidoMaterno + ", email=" + email + ", telefono=" + telefono + '}';
}
```

CURSO DE JAVA EE

www.globalmentoring.com.mx

PASO 9. MODIFICAMOS UN ARCHIVO JAVA

Modificamos el archivo web.xml. Configuramos el Servlet de Jersey, agregando la siguiente configuración:

```
<servlet>
  <servlet-name>JerseyWebApplication</servlet-name>
  <servlet-class>
    org.glassfish.jersey.servlet.ServletContainer
  </servlet-class>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>mx.com.gm.sga.servicio.rest</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>JerseyWebApplication</servlet-name>
  <url-pattern>/webservice/*</url-pattern>
</servlet-mapping>
```

PASO 9. MODIFICAMOS EL CÓDIGO

Archivo web.xml:

Dar click para ir al código

```
<web-app 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-app_3_1.xsd"
  version="3.1">
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <context-param>
    <param-name>javax.faces.FACELETS_REFRESH_PERIOD</param-name>
    <param-value>0</param-value>
  </context-param>
  <context-param>
    <param-name>primefaces.THEME</param-name>
    <param-value>cupertino</param-value>
  </context-param>
  <error-page>
    <exception-type>javax.faces.application.ViewExpiredException</exception-type>
    <location>/faces/index.xhtml</location>
  </error-page>
  <welcome-file-list>
    <welcome-file>faces/index.xhtml</welcome-file>
  </welcome-file-list>
```


PASO 9. MODIFICAMOS EL CÓDIGO

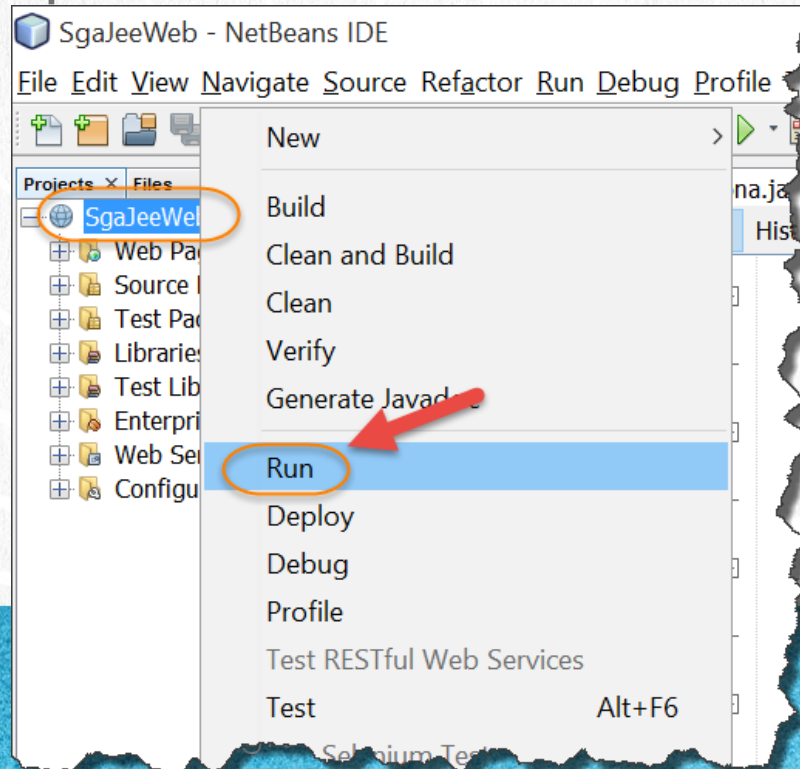
Archivo web.xml:

Dar click para ir al código

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.faces</url-pattern>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>JerseyWebApplication</servlet-name>
  <servlet-class>
    org.glassfish.jersey.servlet.ServletContainer
  </servlet-class>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>mx.com.gm.sga.servicio.rest</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>JerseyWebApplication</servlet-name>
  <url-pattern>/webresource/*</url-pattern>
</servlet-mapping>
</web-app>
```

PASO 10. DESPLIEGUE EN GLASSFISH

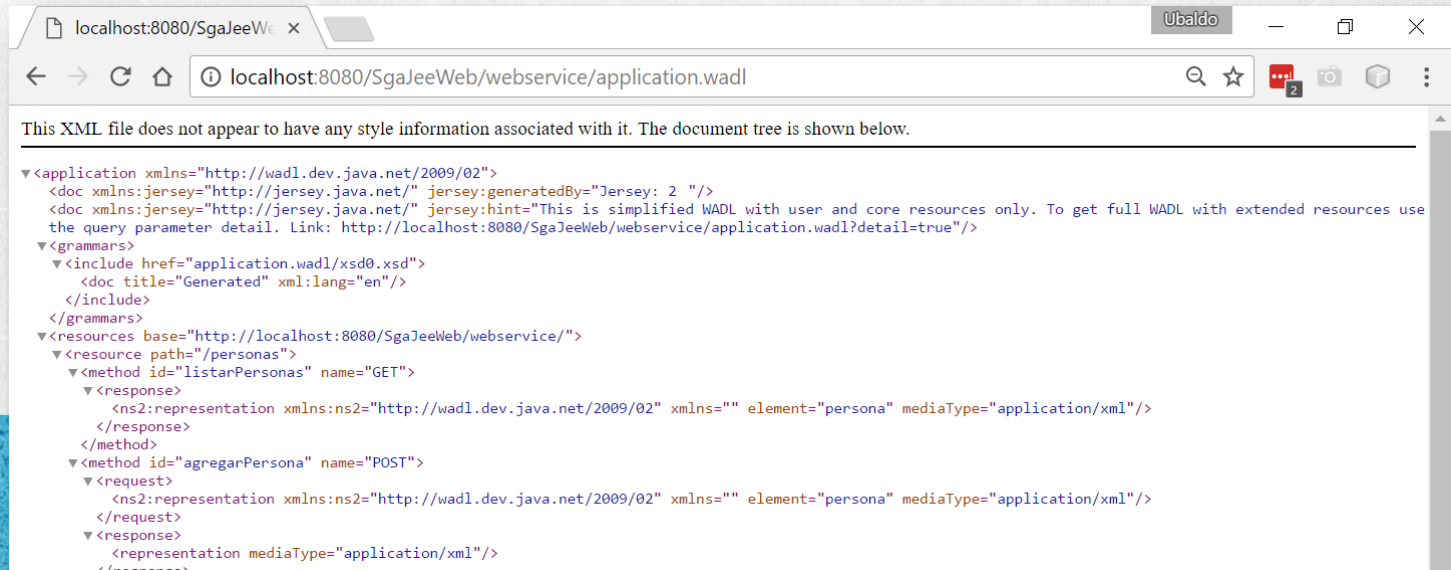
Ejecutamos la aplicación, y esto hará en automático un despliegue de la aplicación:



PASO 11. REVISIÓN DEL REST WEB SERVICE

Ejecutamos la aplicación. Verificamos que se haya desplegado el servicio web rest como la url como sigue:

<http://localhost:8080/SgaJeeWeb/webservice/application.wadl>



The screenshot shows a web browser window with the address bar displaying `localhost:8080/SgaJeeWeb/webservice/application.wadl`. The page content shows the XML document tree for the WADL file. The XML structure is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://wadl.dev.java.net/2009/02">
  <doc xmlns:jersey="http://jersey.java.net/" jersey:generatedBy="Jersey: 2 "/>
  <doc xmlns:jersey="http://jersey.java.net/" jersey:hint="This is simplified WADL with user and core resources only. To get full WADL with extended resources use the query parameter detail. Link: http://localhost:8080/SgaJeeWeb/webservice/application.wadl?detail=true"/>
  <grammars>
    <include href="application.wadl/xsd0.xsd">
      <doc title="Generated" xml:lang="en"/>
    </include>
  </grammars>
  <resources base="http://localhost:8080/SgaJeeWeb/webservice/">
    <resource path="/personas">
      <method id="listarPersonas" name="GET">
        <response>
          <ns2:representation xmlns:ns2="http://wadl.dev.java.net/2009/02" xmlns="" element="persona" mediaType="application/xml"/>
        </response>
      </method>
      <method id="agregarPersona" name="POST">
        <request>
          <ns2:representation xmlns:ns2="http://wadl.dev.java.net/2009/02" xmlns="" element="persona" mediaType="application/xml"/>
        </request>
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
    </resource>
  </resources>
</application>
```


PASO 11. REVISIÓN DEL REST WEB SERVICE

Con el siguiente URL podemos verificar el XSD del Rest Web Service:

<http://localhost:8080/SgaJeeWeb/webservice/application.wadl/xsd0.xsd>



PASO 11. REVISIÓN DEL REST WEB SERVICE

También es posible revisar directamente desde el navegador Web, alguno de los servicios Web publicados. Por ejemplo proporcionando un id_persona válido cualquiera que este sea:

<http://localhost:8080/SgaJeeWeb/webservice/personas/24>



CURSO JAVA EE

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

Con este ejercicio hemos publicado los métodos del EJB utilizando REST Web Services.

Creamos la clase que expone los métodos, así como realizamos las modificaciones respectivas, tanto en la clase de Entidad respectiva, así como en el archivo de configuración web.xml

En el siguiente ejercicio crearemos el cliente que consumirá los web services REST expuestos en este ejercicio.

CURSO ONLINE

JAVA EMPRESARIAL JAVA EE

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



CURSO JAVA EE

www.globalmentoring.com.mx