

**CURSO DE SERVLETS Y JSPS**

# **EJERCICIO**

## **CABECEROS DE RESPUESTA**



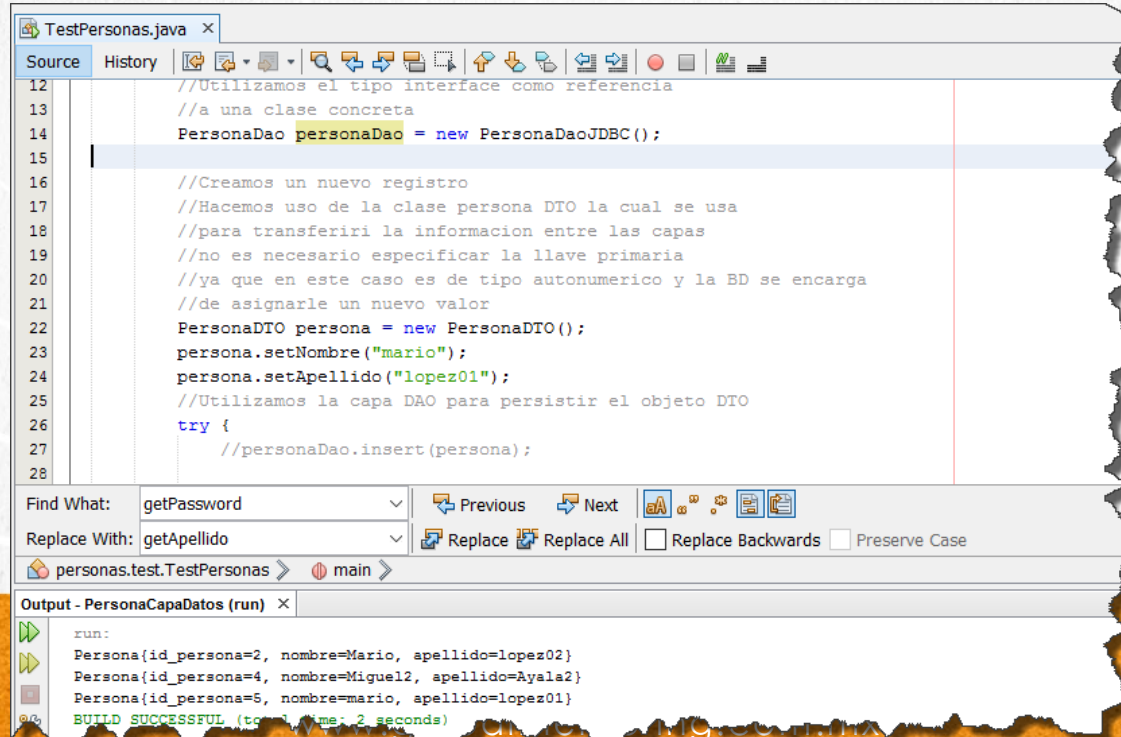
Experiencia y Conocimiento para tu vida

**CURSO DE SERVLETS Y JSPS**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# OBJETIVO DEL EJERCICIO

Crear un programa para practicar los cabeceros de respuesta con Servlets. Al finalizar deberemos observar lo siguiente:



```
TestPersonas.java x
Source History
12 //Utilizamos el tipo interface como referencia
13 //a una clase concreta
14 PersonaDao personaDao = new PersonaDaoJDBC();
15
16 //Creamos un nuevo registro
17 //Hacemos uso de la clase persona DTO la cual se usa
18 //para transferir la informacion entre las capas
19 //no es necesario especificar la llave primaria
20 //ya que en este caso es de tipo autonumerico y la BD se encarga
21 //de asignarle un nuevo valor
22 PersonaDTO persona = new PersonaDTO();
23 persona.setNombre("mario");
24 persona.setApellido("lopez01");
25 //Utilizamos la capa DAO para persistir el objeto DTO
26 try {
27     //personaDao.insert(persona);
28
Find What: getPassword
Replace With: getApellido
Previous Next
Replace Replace All
Replace Backwards Preserve Case
personas.test.TestPersonas main
Output - PersonaCapaDatos (run) x
run:
Persona{id_persona=2, nombre=Mario, apellido=lopez02}
Persona{id_persona=4, nombre=Miguel2, apellido=Ayala2}
Persona{id_persona=5, nombre=mario, apellido=lopez01}
BUILD SUCCESSFUL (total time: 2 seconds)
```

# PASO 1. CREACIÓN DEL PROYECTO

Vamos a crear el proyecto:

New Web Application

**Steps**

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

**Name and Location**

Project Name: CabecerosRespuesta

Project Location: C:\Cursos\ServletsJSPs\Leccion08 Browse...

Project Folder: C:\Cursos\ServletsJSPs\Leccion08\CabecerosRespuesta

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

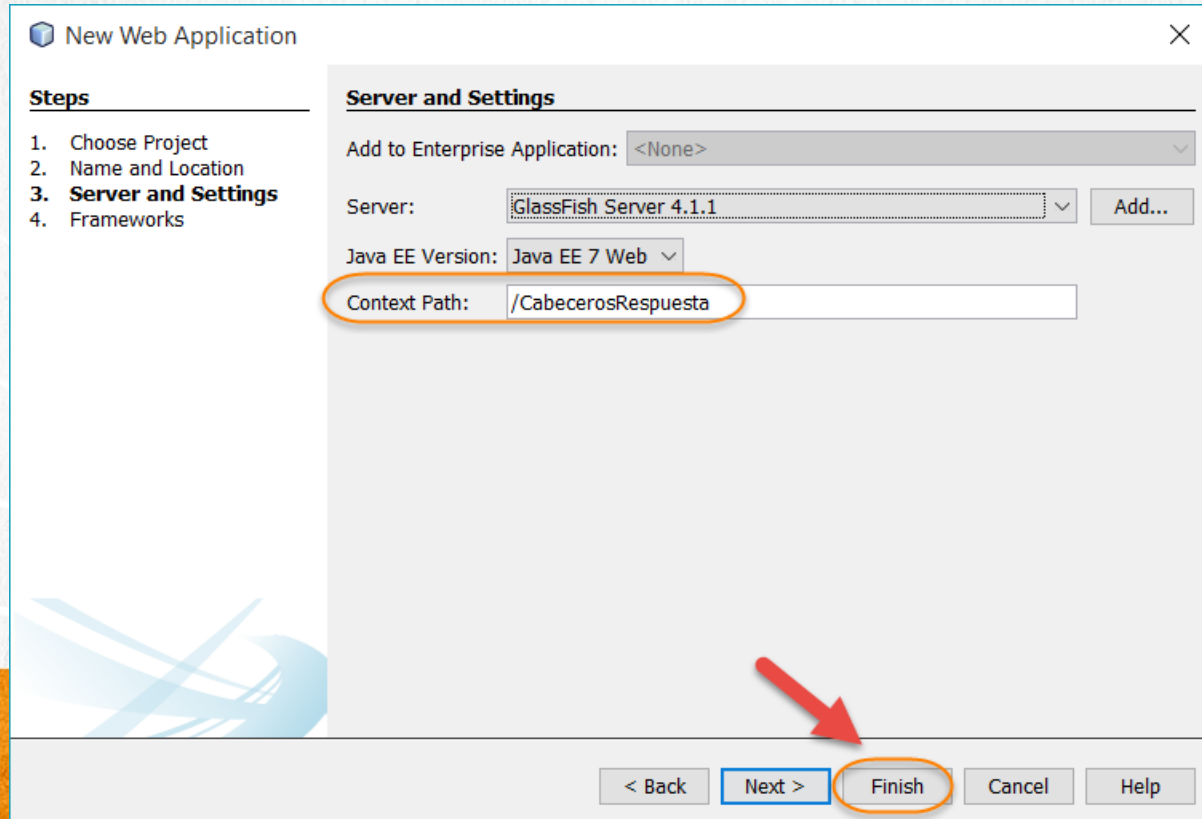
Different users and projects can share the same compilation libraries (see Help for details).

< Back Next > Finish Cancel Help



# PASO 1. CREACIÓN DEL PROYECTO

Vamos a crear el proyecto:



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: /CabecerosRespuesta

< Back Next > Finish Cancel Help

# PASO 2. AGREGAMOS UNA CLASE

Agregamos un Servlet al proyecto:

**New Servlet**

**Steps**

1. Choose File Type
- 2. Name and Location**
3. Configure Servlet Deployment

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

# PASO 3. AGREGAMOS UNA CLASE

Agregamos una clase al proyecto:

**New Servlet**

**Steps**

1. Choose File Type
- 2. Name and Location**
3. Configure Servlet Deployment

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

# PASO 4. MODIFICAMOS EL CÓDIGO

## Archivo index.java:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Ejemplo Cabeceros de Respuesta</title>
  </head>
  <body>
    <h1>Ejemplo Cabeceros de Respuesta</h1>
    <a href="/CabecerosRespuesta/GeneracionExcelServlet">Link al Servlet que genera un excel</a>
    <br>
    <a href="/CabecerosRespuesta/FechaServlet">Link al Servlet que actualiza la fecha</a>
  </body>
</html>
```

# PASO 5. MODIFICAMOS EL CÓDIGO

## Archivo GeneracionExcelServlet.java:

```
package servlets;

import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet(name = "GeneracionExcelServlet", urlPatterns = {"/GeneracionExcelServlet"})
public class GeneracionExcelServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        try (PrintWriter out = response.getWriter()) {
            //Especificamos el tipo del documento de respuesta (MIME)
            response.setContentType("application/vnd.ms-excel");
            response.setHeader("Content-Disposition", "attachment; filename=excelEjemplo.xls");
            //Para un uso más profesional de excel usar: https://poi.apache.org/
            response.setHeader("Pragma", "no-cache");
            response.setHeader("Cache-Control", "no-store");
            response.setDateHeader("Expires", -1);
        }
    }
}
```



# PASO 5. MODIFICAMOS EL CÓDIGO

## Archivo GeneracionExcelServlet.java:

```
//Desplegamos el contenido en un excel
//simplemente modificando el contentType
out.println("\tValores");
out.println("\t1");
out.println("\t2");
out.println("Total\t=SUMA (B2:B3) ");
    }
}
```

# PASO 6. MODIFICAMOS EL CÓDIGO

## Archivo FechaServlet.java:

```
package servlets;

import java.io.*;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet(name = "FechaServlet", urlPatterns = {"/FechaServlet"})
public class FechaServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try (PrintWriter out = response.getWriter()) {
            response.setContentType("text/html;charset=UTF-8");
            response.setHeader("refresh", "1");//dado en segundos
            //Obtenemos la fecha actual y le aplicamos un formato
            Date fecha = new Date();
            //API para utilizar la clase SimpleDateFormat
            //https://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html
```

# PASO 6. MODIFICAMOS EL CÓDIGO

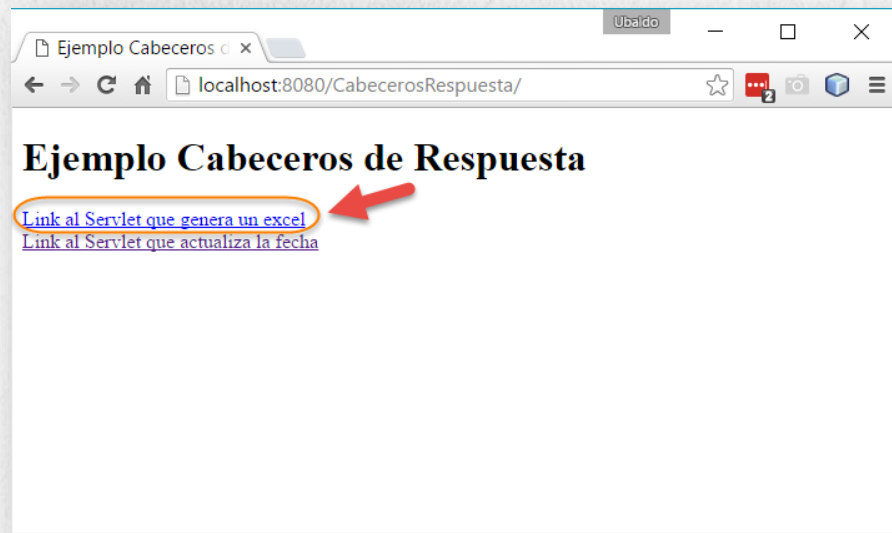
## Archivo FechaServlet.java:

```
//SimpleDateFormat formateador = new SimpleDateFormat("dd 'de' MMMM 'de' yyyy 'a las' HH:mm:ss");
SimpleDateFormat formateador = new SimpleDateFormat("'Hora actualizada' HH:mm:ss");
String fechaConFormato = formateador.format( fecha );
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet FechaServlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Fecha actualizada: " + fechaConFormato + "</h1>");
out.println("</body>");
out.println("</html>");

    }

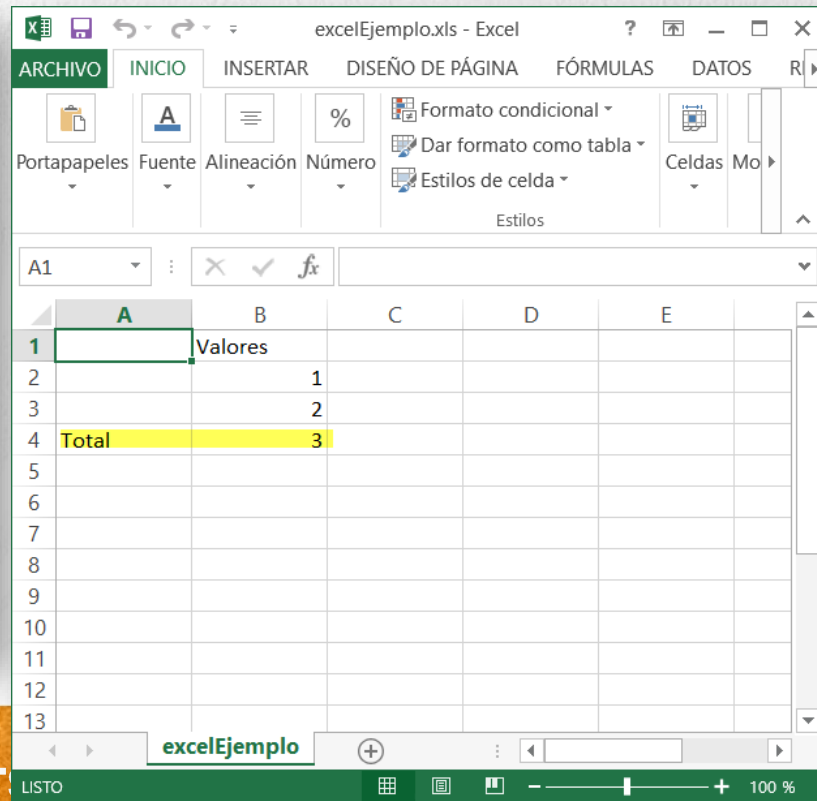
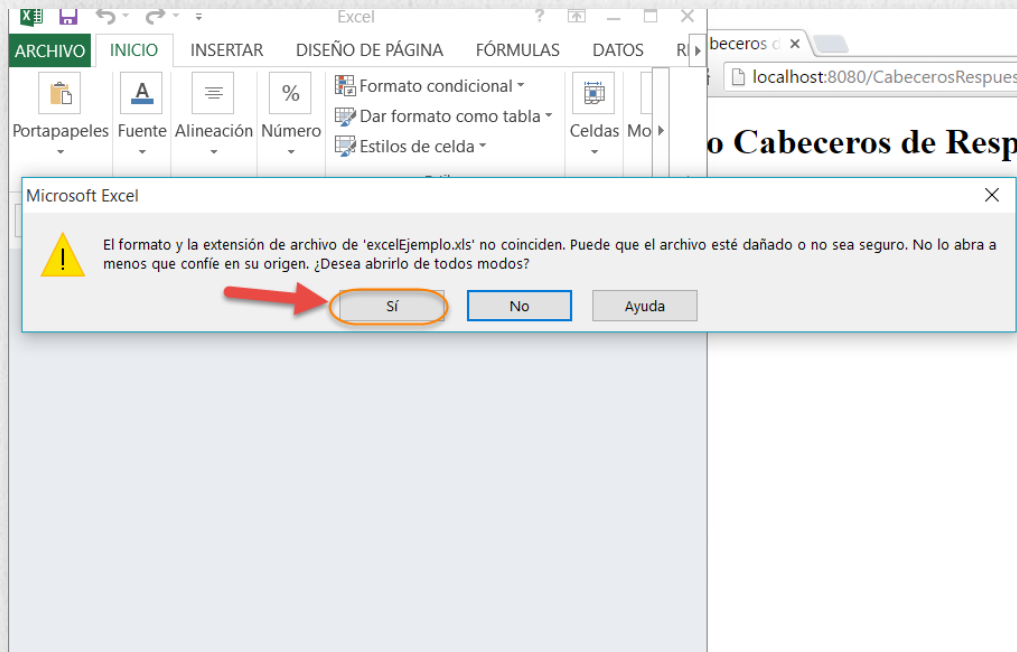
}
```

# PASO 7. EJECUTAMOS EL PROYECTO





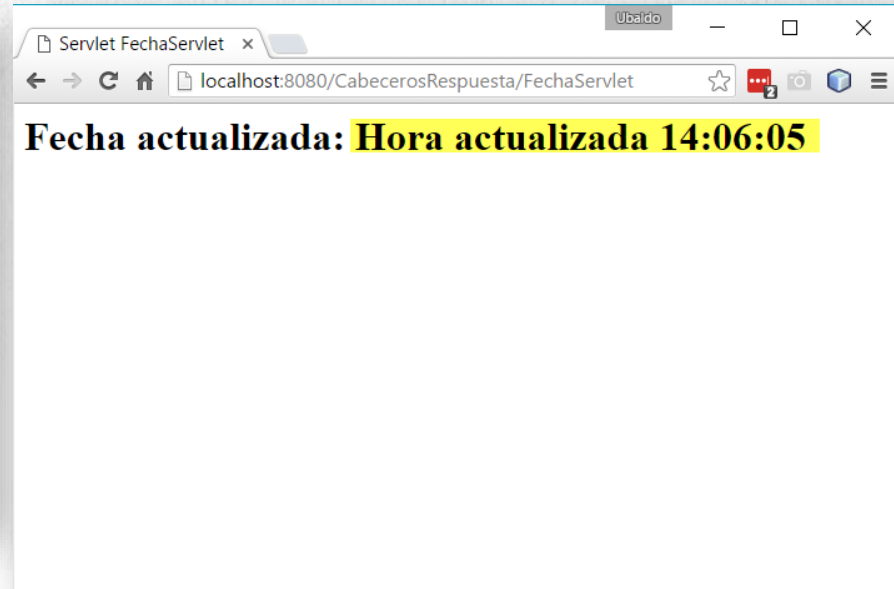
# PASO 7. EJECUTAMOS EL PROYECTO



**CURSO DE SERVLET**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# PASO 7. EJECUTAMOS EL PROYECTO



# CONCLUSIÓN DEL EJERCICIO

Como este ejercicio hemos visto como utilizar algunos cabeceros de respuesta con el API de los Servlets.

Existen muchos usos, pero con los ejemplos que vimos podemos entender a más detalle para qué son y cómo utilizar varios de ellos según el requerimiento que tengamos que desarrollar.



Experiencia y Conocimiento para tu vida

**CURSO DE SERVLETS Y JSPS**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



**CURSO ONLINE**

# **SERVLETS & JSP**

---

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

**CURSO DE SERVLETS Y JSPS**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)