

ENTREGA 1

Diseño y Pruebas



Universidad de Sevilla

Grupo:

José Ángel Domínguez Espinaco

Daniel Lozano Portillo

María Ruiz Gutiérrez

Tania Salguero Álvarez

Miguel Ternero Algarín

Laura Vera Recacha

Contenido

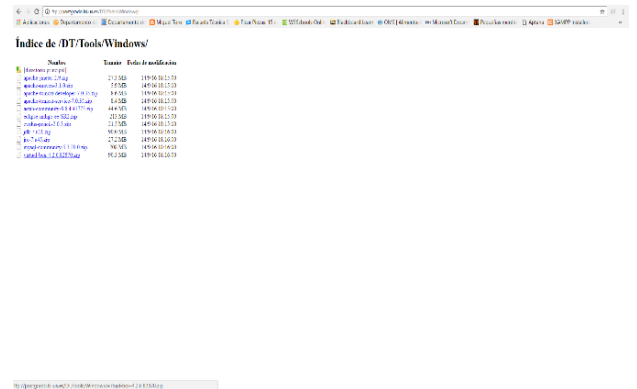
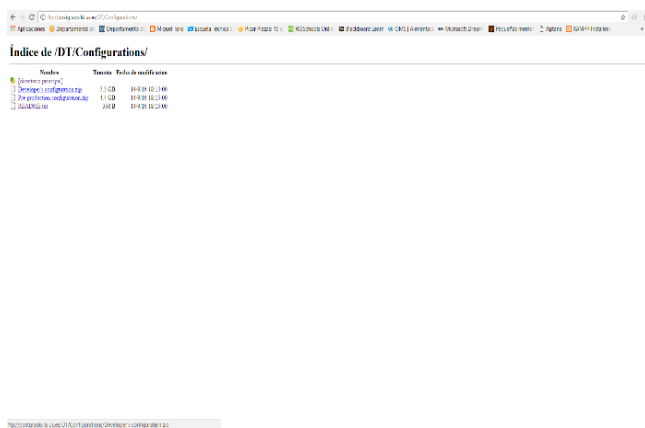
1. Introducción	3
2. Entorno de trabajo	3
3. Requisitos nivel c	13
4. Requisitos nivel b	14
5. Lista de tareas	18
6. Webgrafía	19

1. Introducción

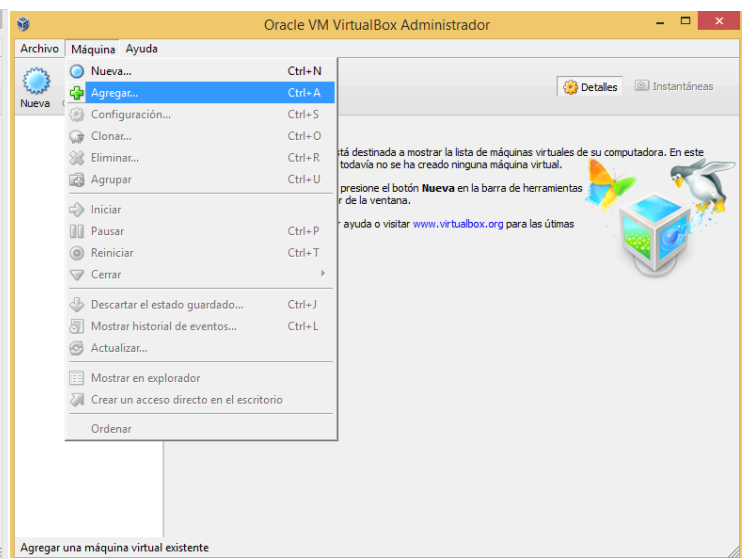
En este documento vamos a describir cómo hemos realizado el proyecto propuesto, así como los errores que nos han ido surgiendo y la solución encontrada. Comenzaremos con una breve descripción de la instalación del entorno de trabajo y seguiremos comentando cómo hemos realizado cada uno de los ítems.

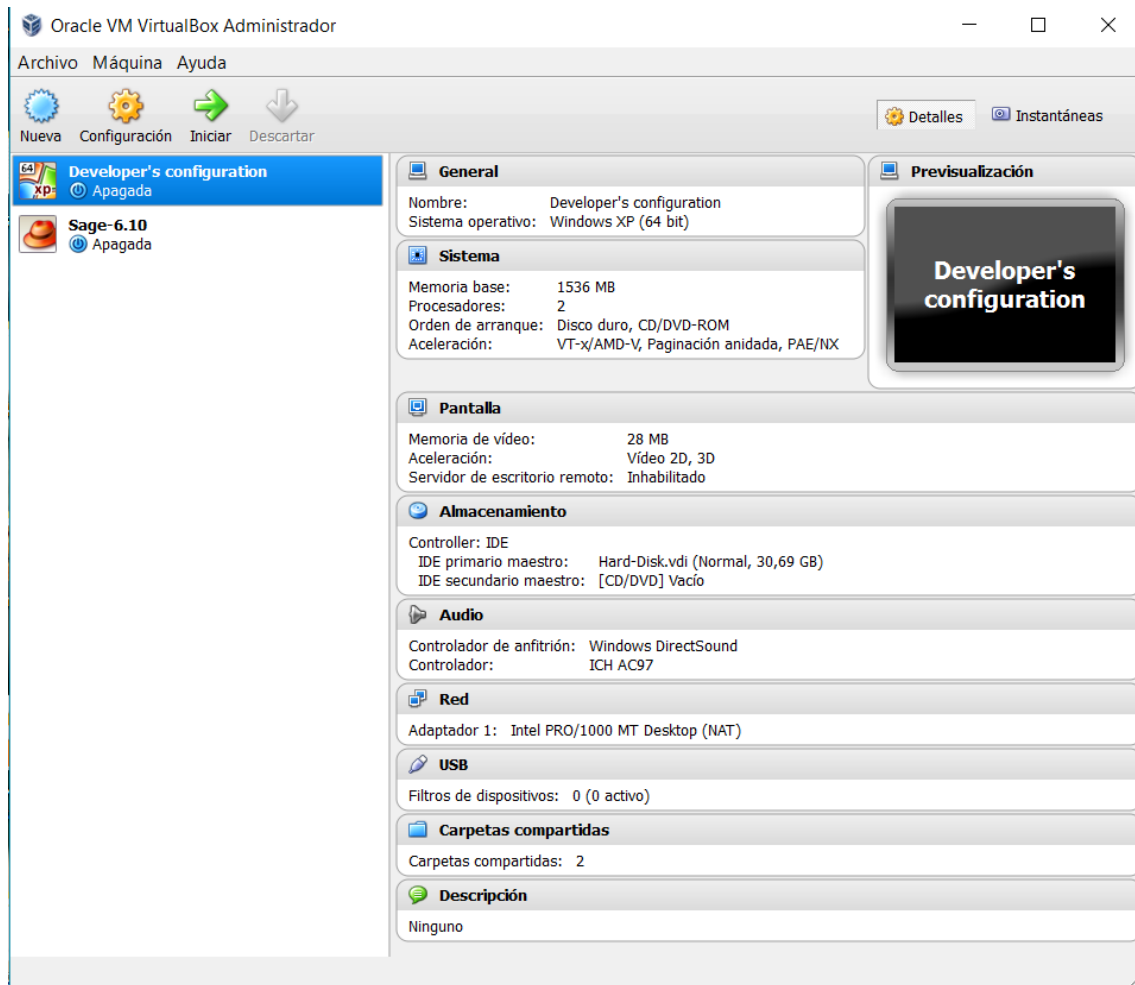
2. Entorno de trabajo

Para montar nuestro propio entorno de trabajo, el cual hemos decidido alojar en una máquina virtual, en primer lugar hemos descargado la imagen del disco necesaria y la máquina virtual del departamento de la asignatura.

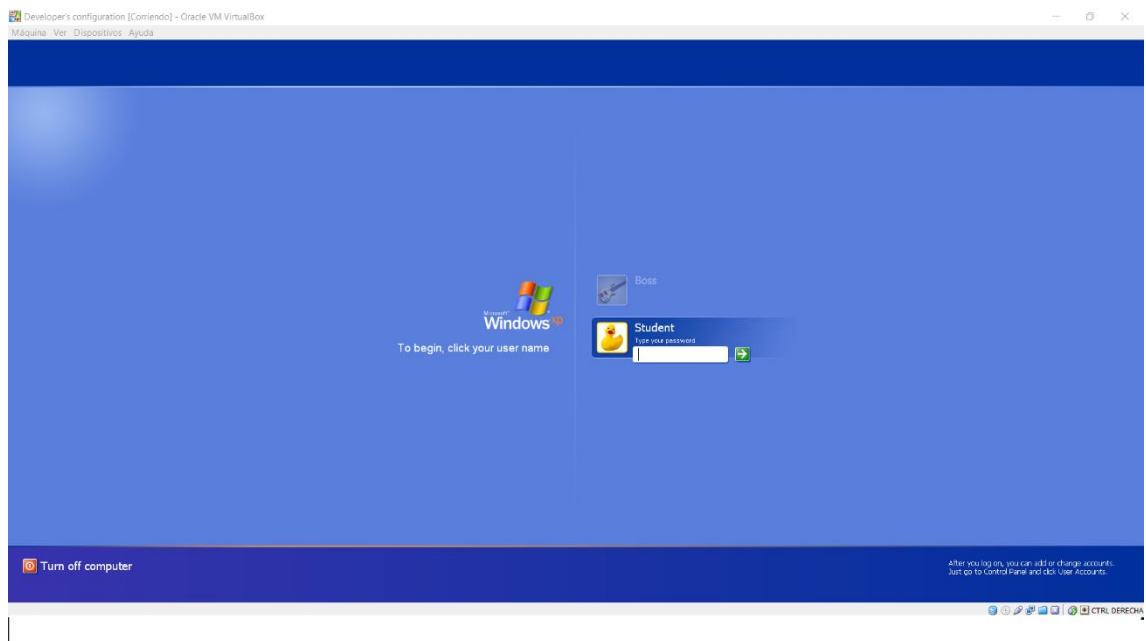


Una vez descargado esto y nuestra máquina virtual en funcionamiento, procedemos a la instalación de la imagen descargada en la misma, configurando los detalles que se nos facilitan sobre el ordenador virtual que estamos recreando.



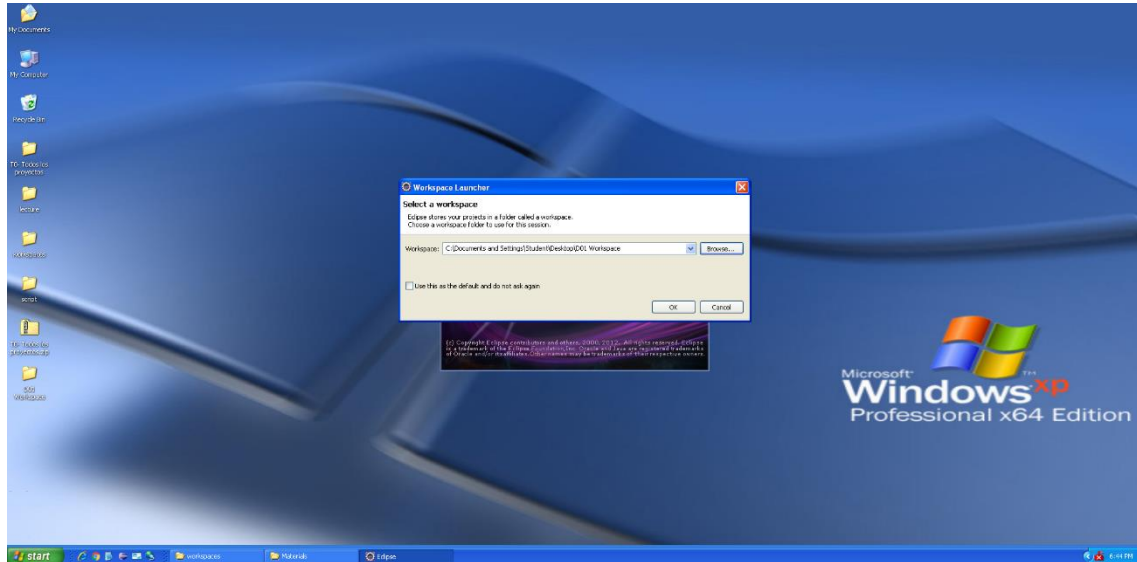


El siguiente paso, es iniciar sesión con el usuario "student" y la contraseña proporcionada en los archivos.

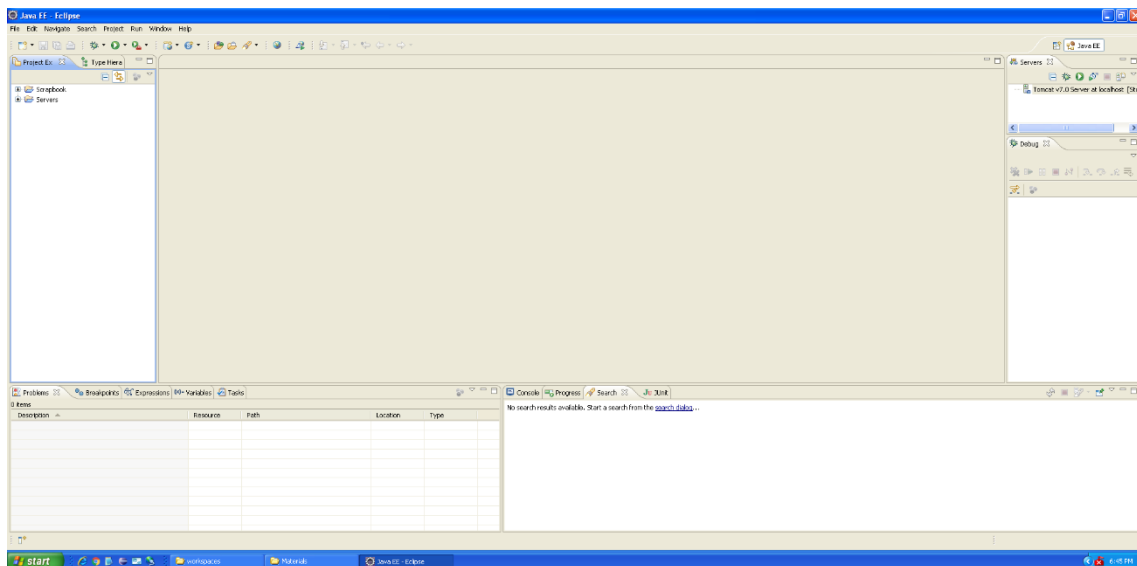


Diseño y Pruebas. Entrega 1

Una vez hemos accedido a dicho usuario, se copia la carpeta llamada "Sample Workspace 1.6" que nos ofrecen en el material de la lección. Se renombramos poniéndole "D01 Workspace". Por último dentro del workspace creado copiamos el proyecto llamado "Sample project 1.6". Al que también renombramos con el nombre de "Sample". Ya tenemos creado nuestro workspace, ahora lo abrimos con eclipse indigo.

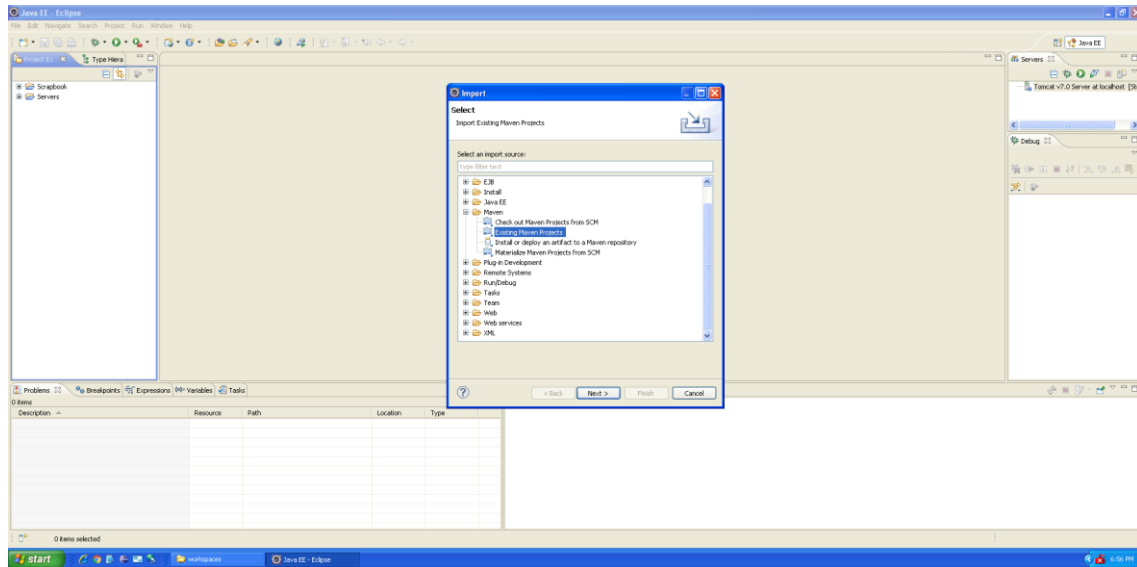


La siguiente vista de eclipse, será la que usaremos en nuestro entorno:

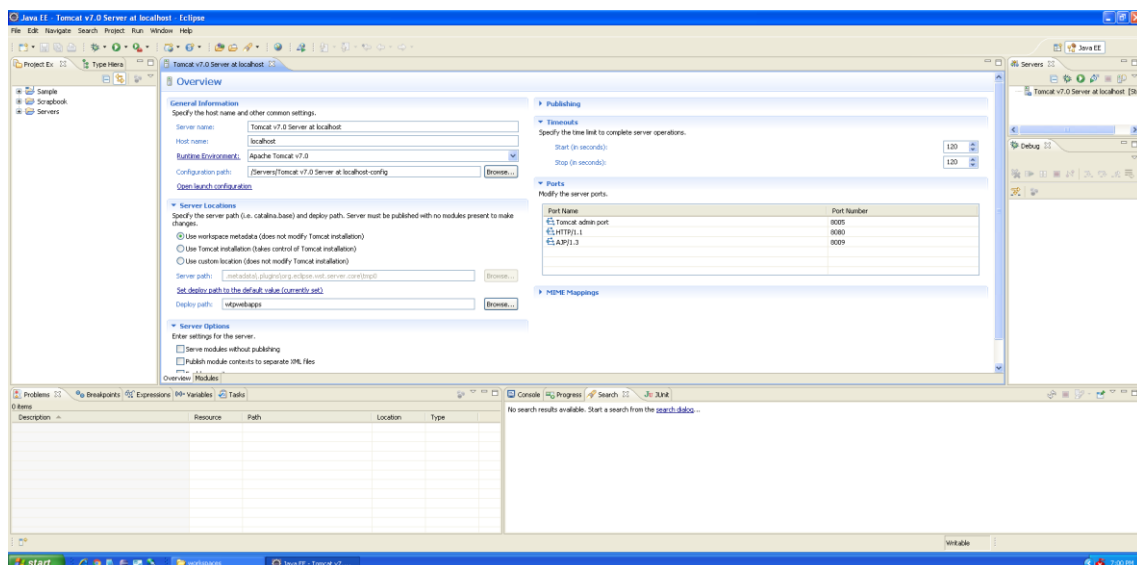


Diseño y Pruebas. Entrega 1

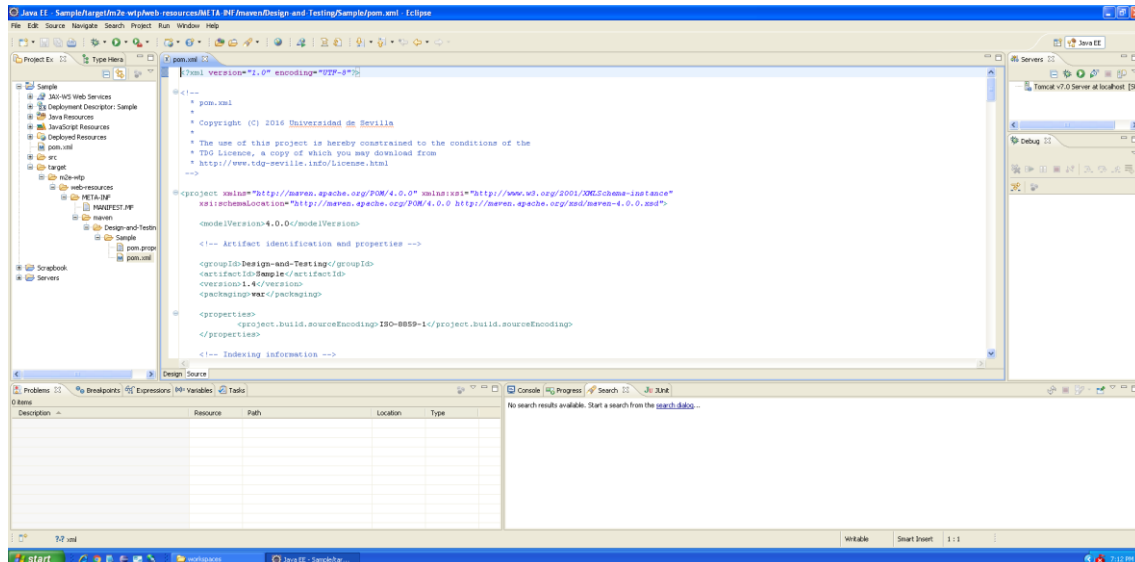
El siguiente paso es importar nuestro proyecto con Maven, para ello vamos a "File -> Import -> MavenExisting Maven Projects", pulsamos en "Browser" y buscamos el archivo llamado "Sample". Seleccionamos el archivo /pom.xml y hacemos clic en "finish".



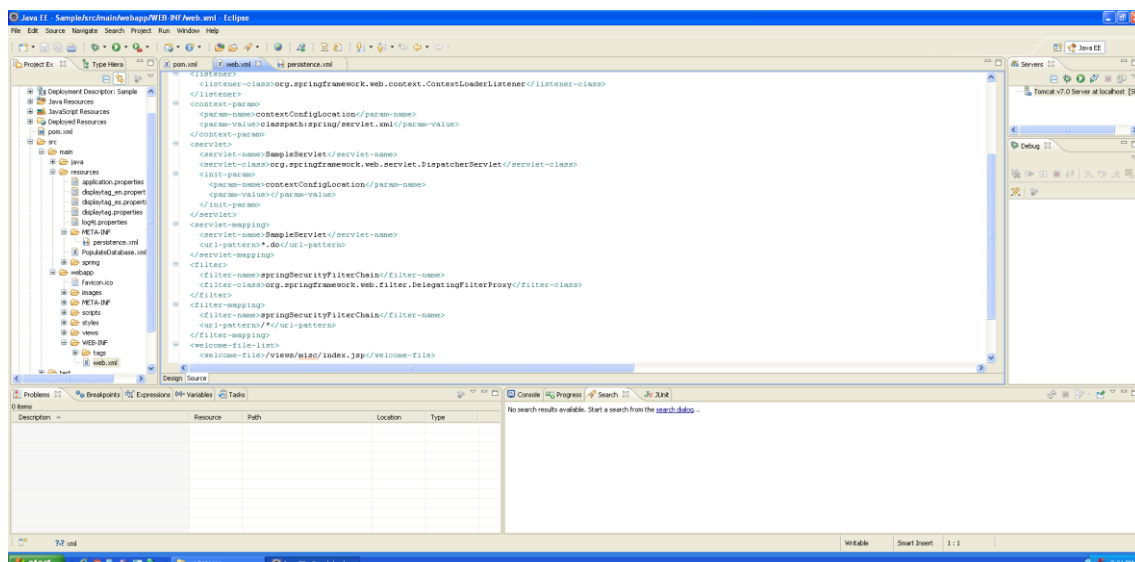
Una vez hecho esto, pasamos a configurar el servidor Tomcat. Para esto, ponemos en HTTP/1.1 el puerto 8080 y cambiamos los tiempos para que el servidor realice las operaciones de arrancar, para ello, le ponemos 120 segundos y para apagarlo 180 segundos.



Ahora, nos toca configurar el proyecto. Abrimos el archivo "pom.xml" y cambiamos los valores de "groupId", "artifactId" y "version". También podemos hacer cambios en "name", "url" y "description". Debemos tener en cuenta que para cada cambio que hagamos en este fichero tenemos que actualizarlo de la siguiente forma: seleccionamos el proyecto con el botón derecho, clicamos en "Maven" y elegimos la opción ""Update project". Posteriormente, seleccionamos "Force update of Snapshots/Releases" y finalmente pulsamos "Ok".

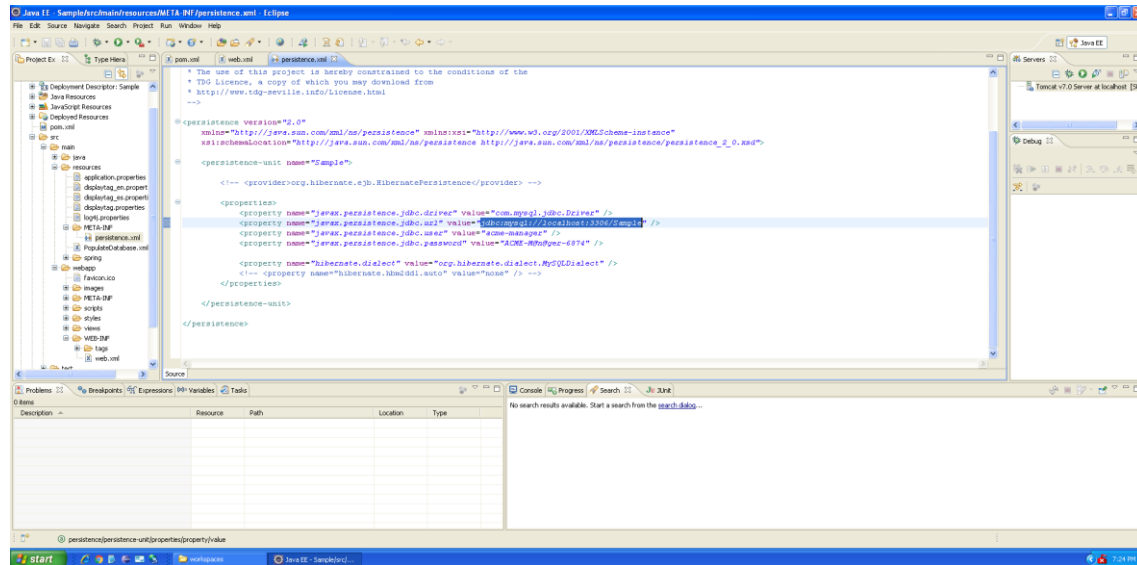


En el siguiente paso, accedemos al archivo "web.xml", en el que tenemos que cambiar el valor de "display-name" (este se refiere al nombre de tu proyecto). También modificamos el campo de "servlet-name", en él se recomienda poner el nombre de tu proyecto con el sufijo "Servlet".

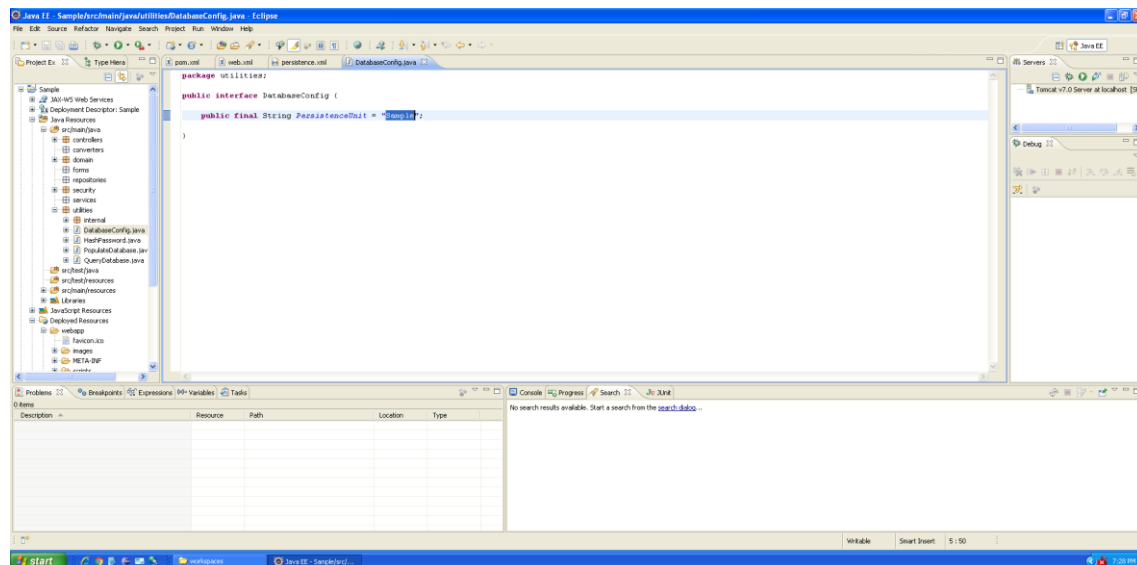


Diseño y Pruebas. Entrega 1

Una vez hecho esto, abrimos el archivo "persistence.xml", donde cambiaremos el valor de "property value" añadiendo el nombre del proyecto al final. Por ejemplo: <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/Sample" />.

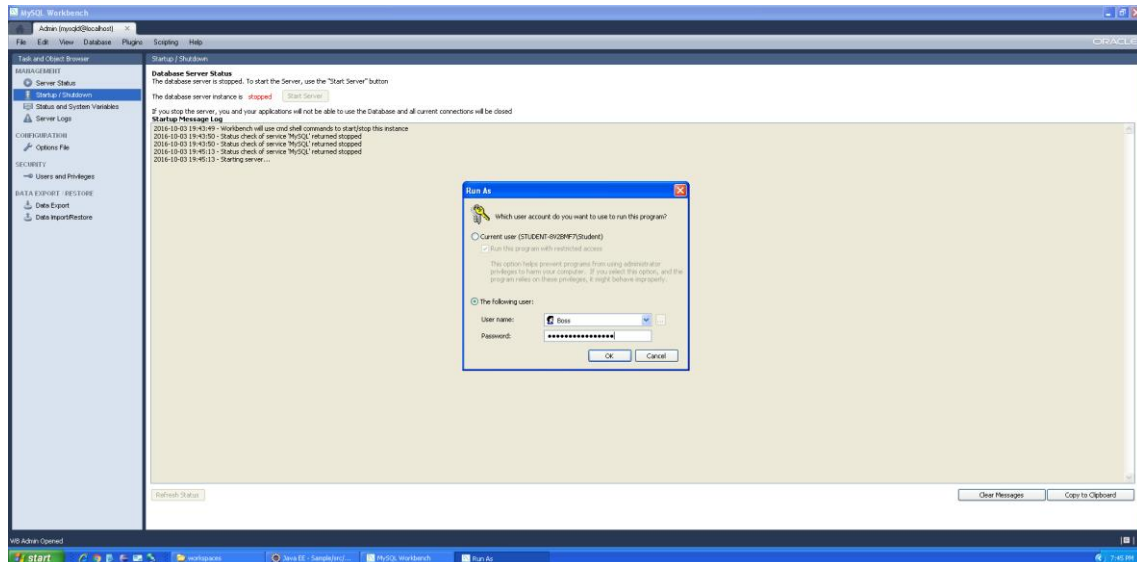


Para configurar la base de datos, abrimos el archivo "DatabaseConfig.java", en el que debemos cambiar el valor del atributo "PersistenceUnit" por el nombre del proyecto.

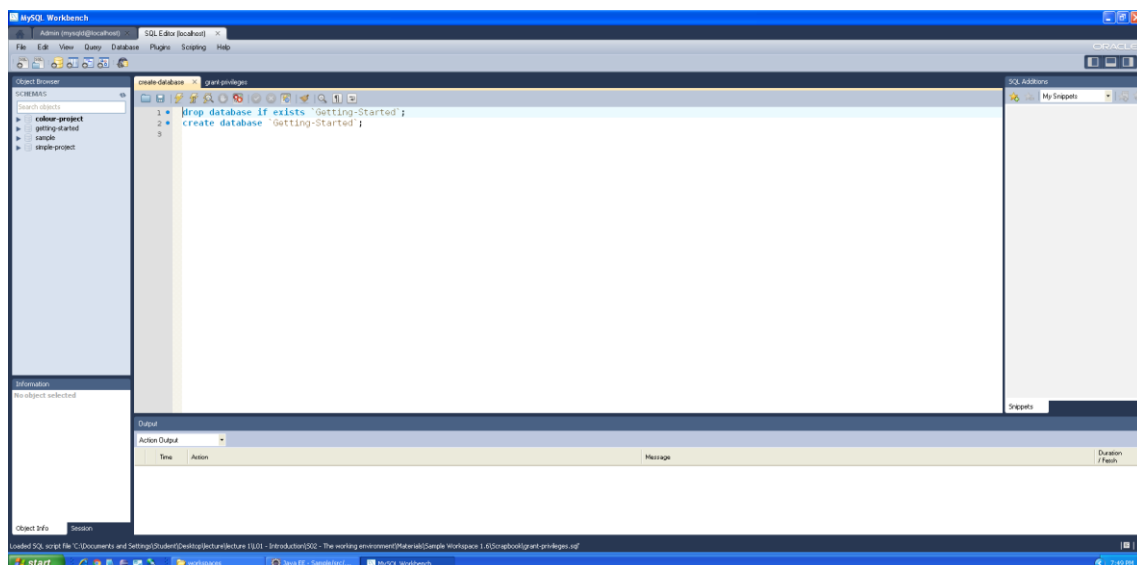


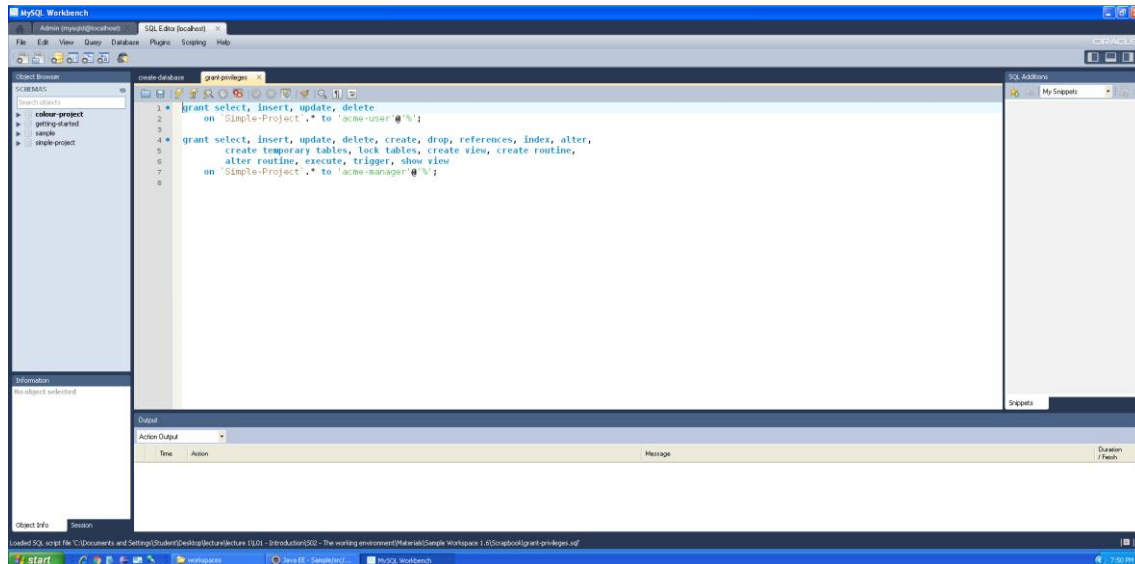
Diseño y Pruebas. Entrega 1

El siguiente paso es iniciar el servidor de la base de datos haciendo clic en "Start Server" y autenticarnos con "boss" cuya contraseña es \$I=B0\$\$=U\$3=P@\$\$. Luego, volvemos a la pantalla inicial, hacemos clic en localhost y abrimos los script que se nos proporcionan en esta lección. El primero de ellos crea la base de datos llamada "Sample" y el segundo, le otorga ciertos privilegios a los usuarios.

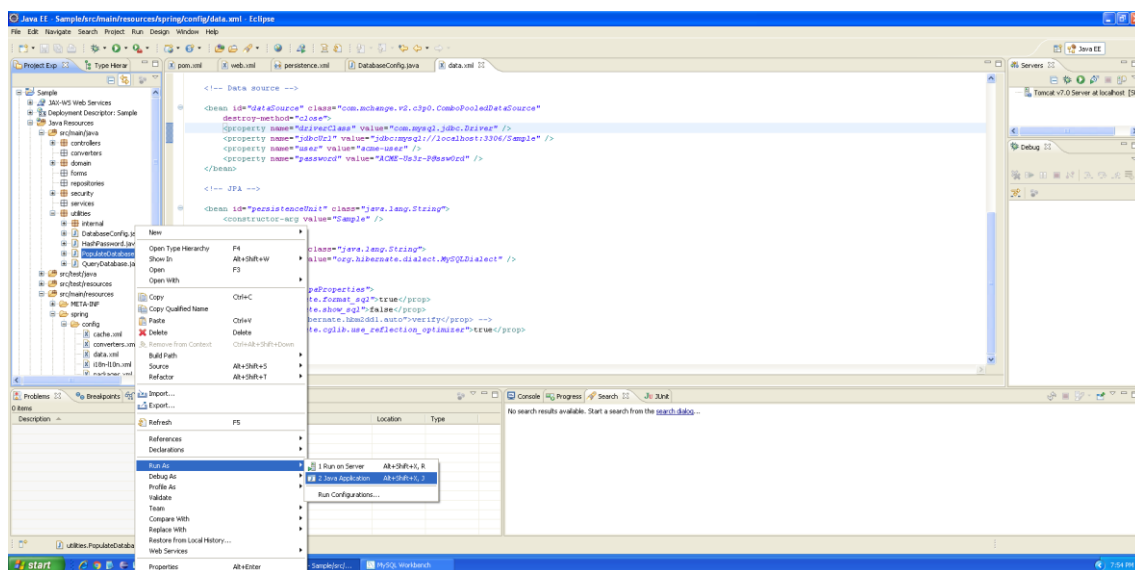


Ahora, procedemos a crear la base de datos y otorgarle privilegios de usuario.

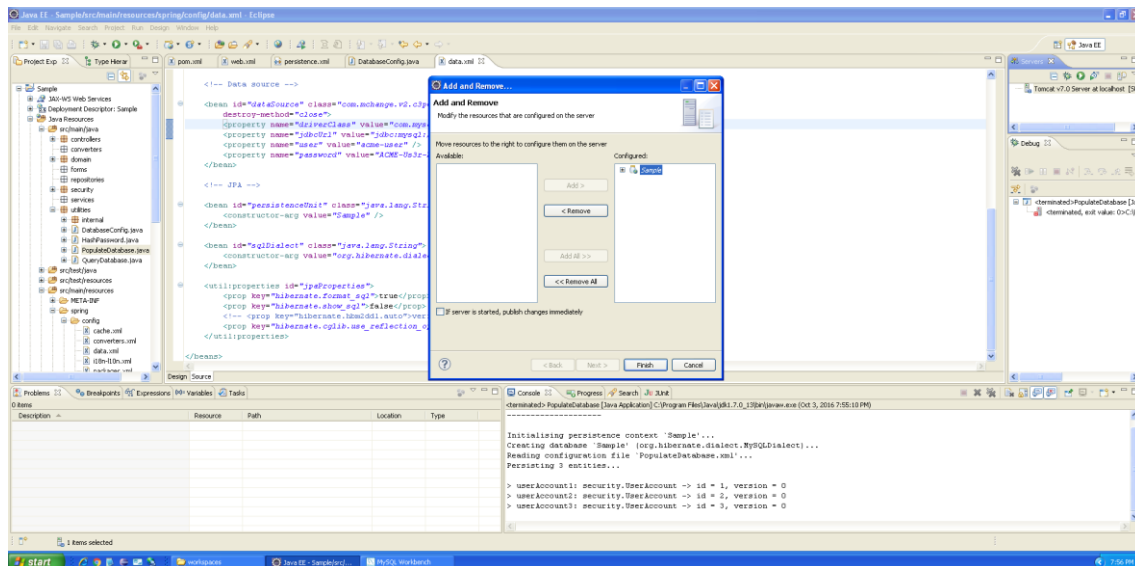




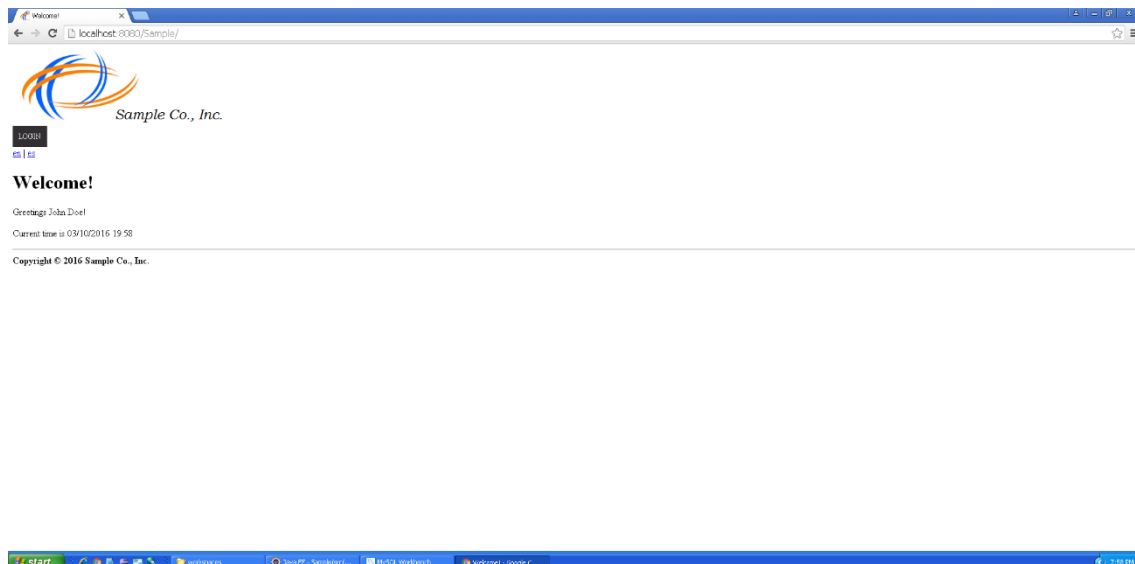
Volvemos al eclipse y ejecutamos el archivo "PopulateDatabase.java" con "Java Application". Esto inserta los datos en la base de datos creada. A continuación añadimos el proyecto al Tomcat haciendo clic derecho sobre "Tomcat 7.0 Server", seleccionando "Add and Remove..." y añadiendo nuestro proyecto. Finalmente clicamos en "Finish" y arrancamos el servidor.



Diseño y Pruebas. Entrega 1



Para finalizar, abrimos nuestro buscador de Google Chrome y accedemos a la siguiente url: <http://localhost:8080/Sample> (nombre de nuestro proyecto).



3. Requisitos nivel c

Este nivel tiene 3 ítems, que continuación vamos a desarrollar.

El primero de ellos, se corresponde con la elaboración de este documento. Para el segundo, Hemos creado un proyecto nuevo tomando como base las plantillas proporcionadas y hemos modificado su pantalla de bienvenida añadiendo los nombres de los integrantes del grupo. Para el último ítem, hemos modificado algunos de los archivos css para modificar el estilo de la página web según nos pedía dicho ítem. Algunos de los cambios que hemos realizado ha sido la modificación del color de fondo de la página, la fuente de la letra... Para ello, en el script llamado common.css cambiamos el color del fondo de la página; también cambiamos el estilo de fuente de la página. Otros cambios realizados en este script son los h1, los cuales hemos cambiado el color de la letra y del fondo. Los cambios realizados son:

```
h1{  
    background-color: gold;  
    color: blueviolet;  
}  
  
body{  
    background-color: floralwhite;  
    font-family:"courier new";  
}
```

En el script llamado jMenu.css hemos realizado el cambio de color de todo el menú y de sus desplegables. Hemos realizado los cambios en .jMenu li, .jMenu li ul li y .jMenu li ul li.arrow; Los cambios quedarían así:

```
.jMenu li {  
    display: table-cell;  
    background-color: indianred;  
    margin: 0px;  
}  
  
.jMenu li ul li {  
    background-color: indianred;  
    display: block;
```

```
border-bottom: 1px solid #484548;
padding: 0;
}
.jMenu li ul li.arrow {
background-color: indianred;
background-image: url('../images/arrow_down.png');
background-repeat: no-repeat;
background-position: center center;
height: 6px;
padding: 0;
```

4. Requisitos nivel b

Este nivel tiene tres subapartados, correspondientes al administrador, el cliente y el perfil.

El primer apartado, consta de dos acciones que se corresponden con el administrador. La primera de ellas consiste en añadir un par de parámetros a su url, y en pantalla se suman ambos números. En la siguiente acción, se muestra una cita al azar, de las que hay ya previamente añadidas. Para ello, hemos hecho los siguientes cambios:

Se añade en la clase action-1.jsp

```
<%
double varx;
double vary;
varx= Double.parseDouble(request.getParameter("x"));
vary= Double.parseDouble(request.getParameter("y"));
out.print(varx + " + " + vary + " = ");
out.print(varx+vary);
%>
```

Para la segunda función, añadimos a la clase action-2.jsp

```
<p><spring:message code="quote.action.2" /></p>
<p><spring:message code="quote1.action.2" /></p>
<p><spring:message code="quote2.action.2" /></p>
<p><spring:message code="quote3.action.2" /></p>
<p><spring:message code="quote4.action.2" /></p>
```

<p><spring:message code="quote5.action.2" /></p>

<p><spring:message code="quote6.action.2" /></p>

<p><spring:message code="quote7.action.2" /></p>

<p><spring:message code="quote8.action.2" /></p>

<p><spring:message code="quote9.action.2" /></p>

Además, se añade en messages.properties lo siguiente

quote.action.2 = I have never let my schooling interfere with my education -- Mark Twain

quote1.action.2 = Live as if you were to die tomorrow. Learn as if you were to live forever -- Mahatma Ghandi

quote2.action.2 = The world is a book and those who do not travel read only one page -- Augustine of Hippo

quote3.action.2 = Education is the most powerful weapon which you can use to change the world -- Nelson Mandela

quote4.action.2 = Whatever the cost of our libraries, the price is cheap compared to that of an ignorant nation -- Walter Cronkite

quote5.action.2 = Only a generation of readers will spawn a generation of writers -- Steven Spielberg

quote6.action.2 = It does not matter how slowly you go as long as you do not stop -- Confucius

quote7.action.2 = Intelligence plus character-that is the goal of true education -- Martin Luther King Jr.

quote8.action.2 = I am not a teacher, but an awakener -- Robert Frost

quote9.action.2 = Do not train a child to learn by force or harshness; but direct them to it by what amuses their minds, so that you may be better able to discover with accuracy the peculiar bent of the genius of each -- Plato

Y por último añadimos en messages.es.properties

quote.action.2 = El software es un gas: se expande hasta llenar su contenedor -- Nathan Myhrvold

quote1.action.2 = Los estándares son siempre obsoletos. Eso es lo que los hace estándares -- Alan Bennett

quote2.action.2 = La física es el sistema operativo del Universo -- Steven R Garman

quote3.action.2 = La imaginación es más importante que el conocimiento. El conocimiento es limitado, mientras que la imaginación no -- Albert Einstein

quote4.action.2 = El mayor enemigo del conocimiento no es la ignorancia, sino la ilusión del conocimiento -- Stephen Hawking

quote5.action.2 = Cuanto más sabes, más te das cuenta de que no sabes nada -- Sócrates

quote6.action.2 = Dime y lo olvido, enséñame y lo recuerdo, involúcrame y lo aprendo -- Benjamín Franklin

quote7.action.2 = El auténtico conocimiento es conocer la extensión de la propia ignorancia -- Confucio

quote8.action.2 = Tus clientes más descontentos son tu mayor fuente de aprendizaje -- Bill Gates

quote9.action.2 = La mejor forma de obtener información correcta de los foros de Usenet es enviar algo incorrecto y esperar las correcciones -- Matthew Austern

En el apartado correspondiente a las funciones del cliente, encontramos dos de ellas. En ambas, se muestra una lista de números; con la diferencia de que en la primera, la lista se corresponde con los senos de los ángulos enteros entre 0,0 y 10,0 radianes; y la segunda, se trata de una lista de coordenadas bidimensionales aleatorias. Para darle funcionalidad, hemos modificado lo siguiente en el código dado:

En la clase action-1.jsp, se añade

```
<%!  
    int x;  
    double res=0;  
    %>  
    <ul>  
    <%  
        for(x=0;x<11;x++){  
            res= Math.sin(x);  
            %>  
            <li><% out.println(res); %></li>  
        } %>  
    </ul>
```

Y en la clase action-2.jsp, se añade

```
<%!  
    int x;  
    double res=0;  
    double res1=0;  
    %>  
    <ul>  
    <%  
        for(x=0;x<11;x++){  
            res1= Math.random()*(10-100)+100;  
            res= Math.random()*(10-100)+100;  
            %>  
            <li><% out.println("(" + res + ", " + res1 + ")"); %></li>  
        } %>  
    </ul>
```


Por otro lado, con respecto al apartado del perfil, encontramos dos acciones. La primera de ellas, muestra dos links, uno correspondiente al lsi y otro al portal de usuarios de la US. La segunda función es un contador de números, que como su propio nombre indica, debe contar hasta el número que insertemos en la url. Se han implementado las acciones de la siguiente forma:

Para la primera, en el fichero action-1.jsp se añade

```
<ul><li><a href="http://www.us.es"><spring:message  
code="profile.action.1.enlace.portal"/></a></li>  
<li><a href="http://www.lsi.us.es"><spring:message  
code="profile.action.1.enlace.departamento"/></a>  
</ul>
```

Además, en el fichero messages_es_propierties se añade

```
profile.action.1.enlace.portal = Universidad de Sevilla  
profile.action.1.enlace.departamento = Dep. de Lenguaje y Sistemas
```

Por último en el fichero messages.properties añadimos

```
profile.action.1.enlace.portal = University of Seville  
profile.action.1.enlace.departamento = Dep. of Software Engineering
```

Para la segunda, se modifica el fichero action-2.jsp añadiendo

```
<%String dato= request.getParameter("count");  
Integer numero= Integer.parseInt(dato);  
out.println("0");  
for(Integer i=1;i<=numero;i++){  
  
    out.println(", "+i);  
}  
%>
```

5. Lista de tareas

Jose Ángel Domínguez Espinaco: -Asistencia a clase 5h 30min

- Instalar todos los programas necesarios 5h
- Estudiar el funcionamiento del framework y componentes 2h
- Unir el código final 2h
- Reunión grupal 4h
- Total: 18h 30min -> 148€

Daniel Lozano Portillo: -Asistencia a clase 5h 30min

- Realización del ítem 4 del primer entregable 3h
- Reunión grupal 4h
- Revisión de la entrega final 1h 30min
- Total: 14h -> 112€

María Ruiz Gutiérrez: -Asistencia a clase 5h 30min

- Realización del ítem 4 del primer entregable 8h
- Reunión grupal 4h
- Total: 17h 30 min -> 140€

Tania Salguero Álvarez: -Asistencia a clase 5h 30min

- Instalar la máquina virtual y el entorno de trabajo 3h
- Realización del ítem 1 del primer entregable 3h
- Reunión grupal 4h
- Total: 15h 30min -> 124€

Miguel Ternero Algarín: -Asistencia a clase 5h 30min

- Realización del ítem 1 del primer entregable 4h
- Revisión de la entrega final 2h
- Reunión grupal 4h
- Total: 15h 30min -> 124€

Laura Vera Recacha: -Asistencia a clase 5h 30min

- Realización del ítem 3 del primer entregable 1h 30min
- Realización del ítem 1 del primer entregable 2h
- Reunión grupal 4h
- Total: 13h -> 104€

Hemos considerado un sueldo mínimo de 8€/hora

6. Webgrafía

<http://www.tusalario.es/main/salario/comparatusalario?job-id=2512010000000#/>