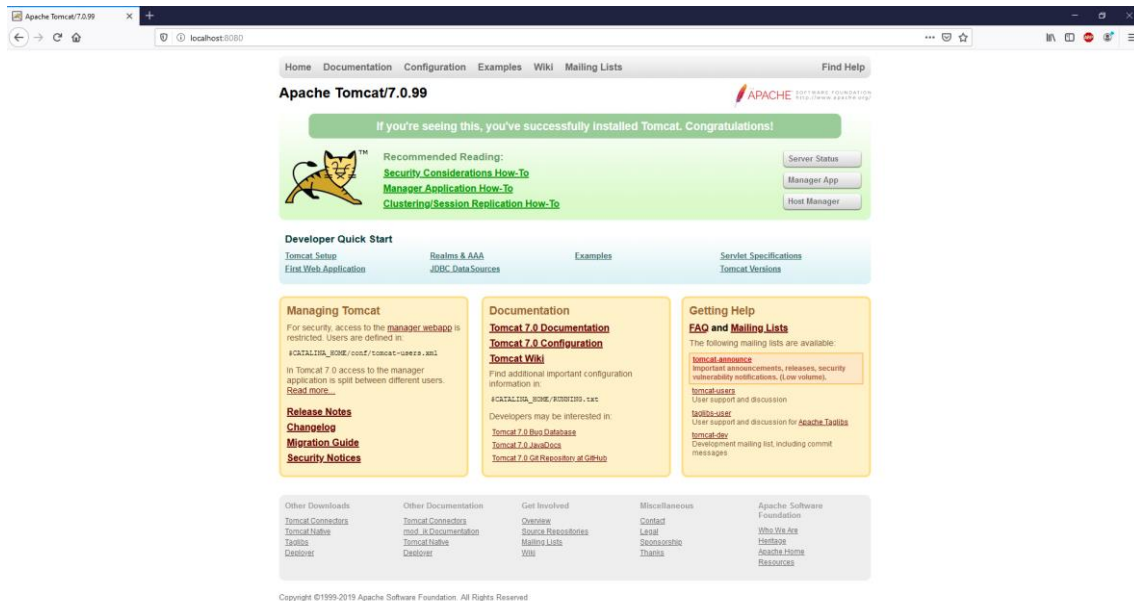


# PRACTICA 1 DE TOMCAT

1. Partimos de que tenemos un servidor Tomcat instalando y funcionando en localhost trabajando por el puerto 8080. Si no lo tenemos debemos revisar las prácticas anteriores donde si instaló y configuró este servicio. Para el comienzo de esta práctica partimos de este punto:



2. Entramos en el manager webapp e introducimos el usuario administrador que tenemos creado:

## Managing Tomcat

For security, access to the **manager webapp** is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 7.0 access to the manager application is split between different users.  
[Read more...](#)

[Release Notes](#)

[Changelog](#)

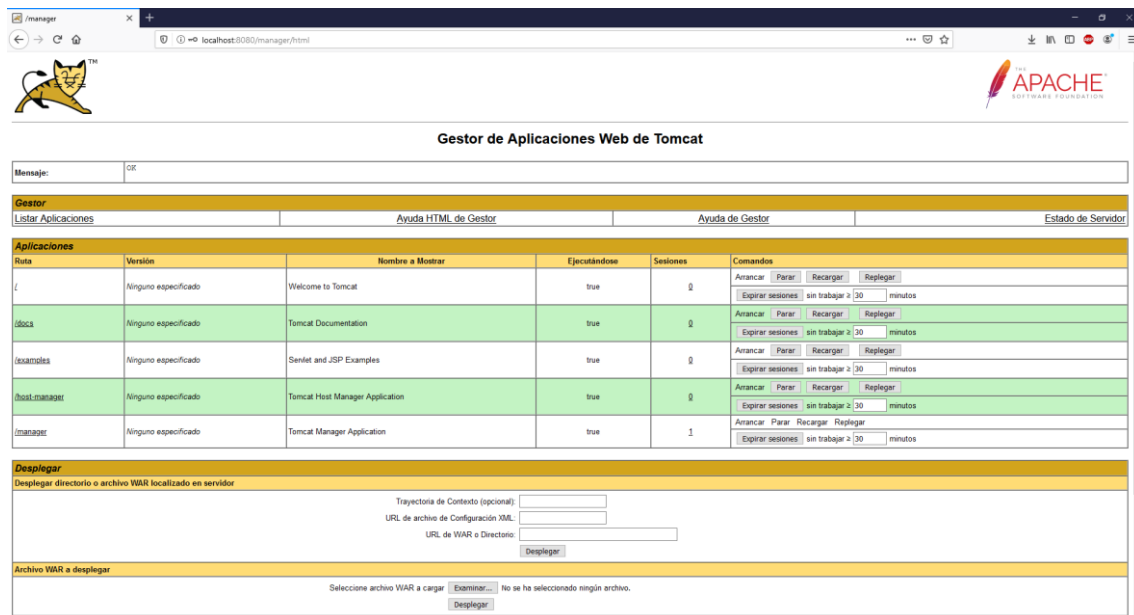
[Migration Guide](#)

[Security Notices](#)

Para ello debemos ir al directorio donde tenemos instalado Tomcat > conf y abrir tomcat-users.xml. Debemos añadir lo siguiente:

```
<role rolename="manager-gui"/>
<user username="admin" password="admin" roles="manager-gui"/>
```

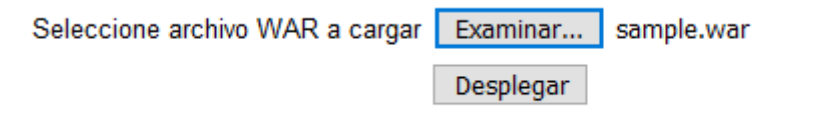
3. En el ejemplo anterior estamos estableciendo el usuario “admin” y la pass “admin”. Podemos cambiarla en ese archivo en cualquier momento. Una vez dentro veremos algo como lo siguiente:



Descárgate la aplicación .war ejemplo de Tomcat de la siguiente URL:

<https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/>

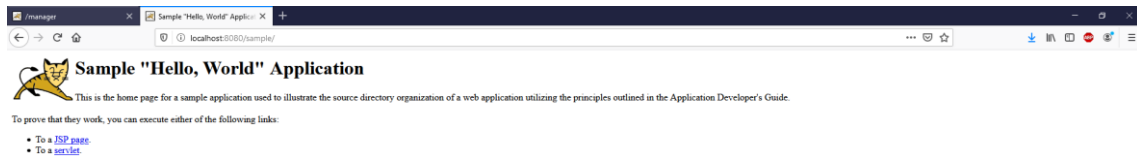
4. Cárgala en tu Tomcat una vez descargada:



5. Ahora veremos como en la parte de aplicaciones está instalada la aplicación de ejemplo

Aplicaciones					
Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	<div> <div>Avanzar</div> <div>Parar</div> <div>Recargar</div> <div>Replegar</div> </div> <div>Expirar sesiones sin trabajar &gt; 30 minutos</div>
/docs	Ninguno especificado	Tomcat Documentation	true	0	<div> <div>Avanzar</div> <div>Parar</div> <div>Recargar</div> <div>Replegar</div> </div> <div>Expirar sesiones sin trabajar &gt; 30 minutos</div>
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	<div> <div>Avanzar</div> <div>Parar</div> <div>Recargar</div> <div>Replegar</div> </div> <div>Expirar sesiones sin trabajar &gt; 30 minutos</div>
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	<div> <div>Avanzar</div> <div>Parar</div> <div>Recargar</div> <div>Replegar</div> </div> <div>Expirar sesiones sin trabajar &gt; 30 minutos</div>
/manager	Ninguno especificado	Tomcat Manager Application	true	1	<div> <div>Avanzar</div> <div>Parar</div> <div>Recargar</div> <div>Replegar</div> </div> <div>Expirar sesiones sin trabajar &gt; 30 minutos</div>
/sample	Ninguno especificado	Hello, World Application	true	0	<div> <div>Avanzar</div> <div>Parar</div> <div>Recargar</div> <div>Replegar</div> </div> <div>Expirar sesiones sin trabajar &gt; 30 minutos</div>

6. Si entramos en la dirección localhost:8080/sample podemos ver el módulo que acabamos de cargar.



7. Ahora copiamos el contenido del módulo de ejemplo y modificamos algunos de sus archivos empezando por el index.html

```

1  <html>
2  <head>
3    <title>Práctica 1 Tomcat</title>
4  </head>
5  <body bgcolor=white>
6
7    <table border="0">
8      <tr>
9        <td>
10       
11     </td>
12     <td>
13       <ul>
14         <li>Ir a<a href="hello.jsp">página JSP</a>.
15         <li>Ir a<a href="hello">tabla del 9</a>.
16       </ul>
17     </td>
18   </tr>
19 </table>
20 </body>
21 </html>

```

8. Modificad también el archivo. jsp para que muestre una imagen propia y vuestro nombre y apellidos

```

1  <html>
2  <head>
3    <title>Álvaro Real</title>
4  </head>
5  <body bgcolor=white>
6
7    <table border="0">
8      <tr>
9        <td align=center>
10         
11       </td>
12      <td>
13        Álvaro Real
14      </td>
15    </tr>
16  </table>
17
18  <%= new String("Álvaro Real") %>
19
20 </body>
21 </html>
22

```

9. Y creamos un archivo .java para contener las instrucciones escritas en lenguaje java para después construir el servlet

```

1  package mypackage;
2
3  import java.io.IOException;
4  import java.io.PrintWriter;
5  import javax.servlet.ServletException;
6  import javax.servlet.http.HttpServlet;
7  import javax.servlet.http.HttpServletRequest;
8  import javax.servlet.http.HttpServletResponse;
9
10 // Extend HttpServlet class
11 public class hello extends HttpServlet {
12
13     private String message;
14
15     public void init() throws ServletException {
16         // Do required initialization
17         message = "Hello World";
18     }
19
20     public void doGet(HttpServletRequest request, HttpServletResponse response)
21         throws ServletException, IOException {
22
23         // Set response content type
24         response.setContentType("text/html");
25
26         // Actual logic goes here.
27         PrintWriter out = response.getWriter();
28         out.println("<h1>" + message + "</h1>");
29         response.setContentType("text/html");
30         response.setCharacterEncoding("UTF-8");
31
32         try (PrintWriter writer = response.getWriter()) {
33             writer.println("<html>");
34             writer.println("<head>");
35             writer.println("<meta charset='UTF-8' />");
36             writer.println("</head>");
37             writer.println("<body>");
38             writer.println("<h1>Tabla del 9</h1>");
39             for (int i = 0; i < 10; i++) {
40                 writer.println("9 x " + i + " = " + 9*i);
41             }
42             writer.println("</body>");
43             writer.println("</html>");
44         }
45     }
46
47     public void destroy() {
48         // do nothing.
49     }
50 }

```








10. Para poder compilar este archivo .java necesitamos el paquete tomcat-api.jar que se encuentra en el servidor en la ruta /usr/share/tomcat7/lib

11. Para compilar el archivo .java basta con hacer:

```
javac -classpath servlet-api.jar hello.java
```

Si no da errores es que lo hemos escrito correctamente.

12. Esta compilación nos deja un archivo .class que es el que debemos incluir en nuestra aplicación:

<input type="checkbox"/> Nombre	Fecha de modificación	Tipo	Tamaño
 images	31/12/2019 20:34	Carpeta de archivos	
 META-INF	31/12/2019 20:28	Carpeta de archivos	
 WEB-INF	31/12/2019 20:28	Carpeta de archivos	
<input checked="" type="checkbox"/>  hello.class	31/12/2019 21:00	Archivo CLASS	1 KB
 hello.java	31/12/2019 21:01	Archivo JAVA	1 KB
 hello.jsp	31/12/2019 20:32	Archivo JSP	1 KB
 index.html	31/12/2019 20:30	Firefox HTML Doc...	1 KB

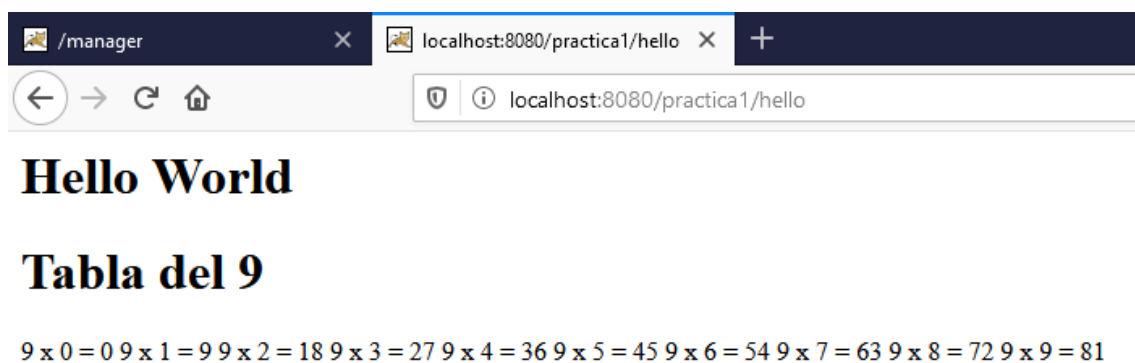
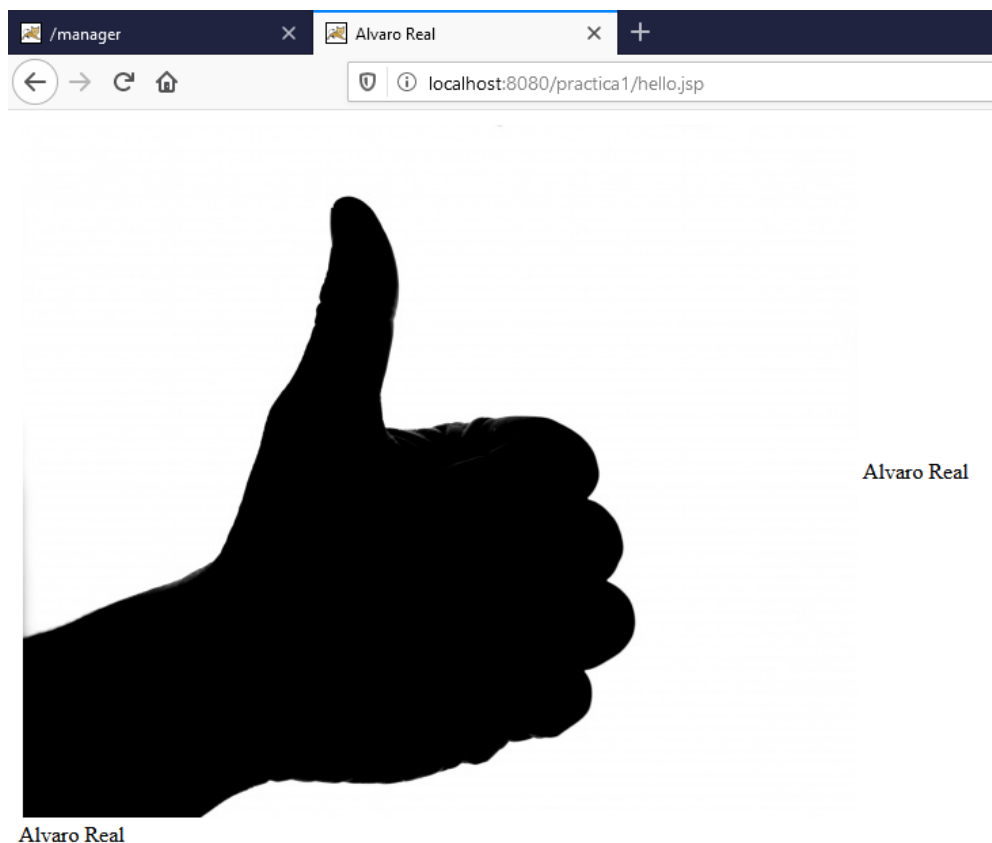
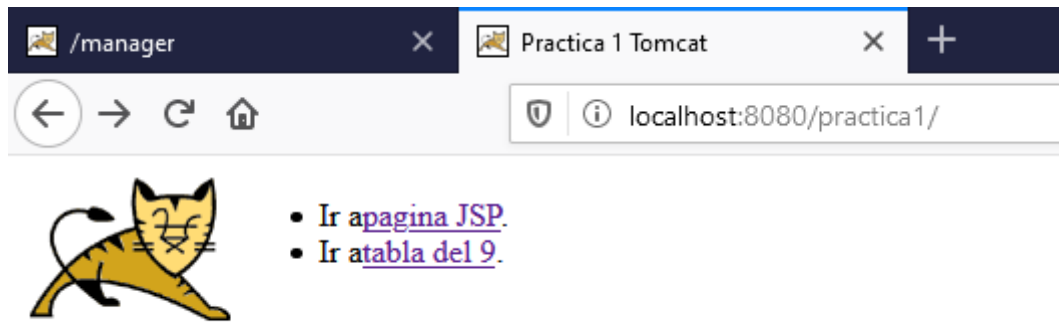
13. Hay que modificar el archivo web.xml para incluir la información del servlet

```
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
5      version="2.4">
6
7      <display-name>Practical</display-name>
8      <description>
9          Práctica 1 - Tomcat - Álvaro Real
10     </description>
11
12     <servlet>
13         <servlet-name>HelloServlet</servlet-name>
14         <servlet-class>mypackage.Hello</servlet-class>
15     </servlet>
16
17     <servlet-mapping>
18         <servlet-name>HelloServlet</servlet-name>
19         <url-pattern>/hello</url-pattern>
20     </servlet-mapping>
21
22 </web-app>
23
```

14. Ahora comprimimos en un archivo practical.war la aplicación para ser desplegado en el servidor tomcat y lo subimos.

```
jar -cvf practical.war .
```

15. Una vez subido lo probados y se deben apreciar nuestros cambios:



Con esta práctica hemos desplegado aplicaciones en Tomcat y hemos configurado un servlet y lo hemos subido.