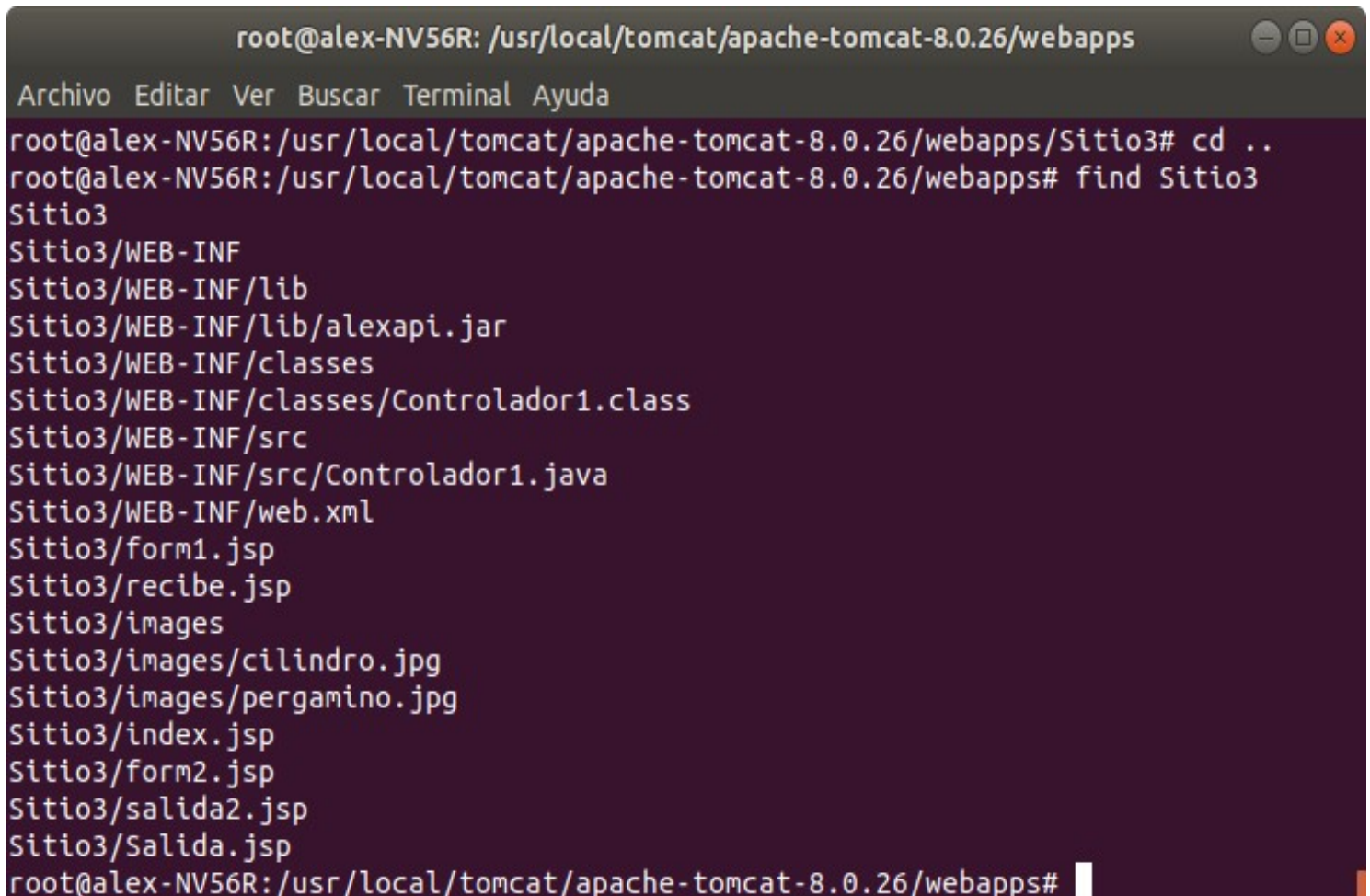


MVC PARTE2 (Cilindro)

Objetivo

Crear un jsp que obtenga datos, después los envíe a un servlet y posteriormente lo muestre en otro jsp, de esta forma podremos ver como funciona el modelo vista controlador

1.-La forma en la que estará conformada nuestro sitio sera la siguiente

A terminal window titled 'root@alex-NV56R: /usr/local/tomcat/apache-tomcat-8.0.26/webapps' with standard window controls. The terminal shows a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The command sequence is: 'cd ..' followed by 'find Sitio3'. The output lists the following files and directories: Sitio3, Sitio3/WEB-INF, Sitio3/WEB-INF/lib, Sitio3/WEB-INF/lib/alexapi.jar, Sitio3/WEB-INF/classes, Sitio3/WEB-INF/classes/Controlador1.class, Sitio3/WEB-INF/src, Sitio3/WEB-INF/src/Controlador1.java, Sitio3/WEB-INF/web.xml, Sitio3/form1.jsp, Sitio3/recibe.jsp, Sitio3/images, Sitio3/images/cilindro.jpg, Sitio3/images/pergamino.jpg, Sitio3/index.jsp, Sitio3/form2.jsp, Sitio3/salida2.jsp, and Sitio3/Salida.jsp. The prompt returns to 'root@alex-NV56R: /usr/local/tomcat/apache-tomcat-8.0.26/webapps#'.

```
root@alex-NV56R: /usr/local/tomcat/apache-tomcat-8.0.26/webapps
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@alex-NV56R:/usr/local/tomcat/apache-tomcat-8.0.26/webapps/Sitio3# cd ..
root@alex-NV56R:/usr/local/tomcat/apache-tomcat-8.0.26/webapps# find Sitio3
Sitio3
Sitio3/WEB-INF
Sitio3/WEB-INF/lib
Sitio3/WEB-INF/lib/alexapi.jar
Sitio3/WEB-INF/classes
Sitio3/WEB-INF/classes/Controlador1.class
Sitio3/WEB-INF/src
Sitio3/WEB-INF/src/Controlador1.java
Sitio3/WEB-INF/web.xml
Sitio3/form1.jsp
Sitio3/recibe.jsp
Sitio3/images
Sitio3/images/cilindro.jpg
Sitio3/images/pergamino.jpg
Sitio3/index.jsp
Sitio3/form2.jsp
Sitio3/salida2.jsp
Sitio3/Salida.jsp
root@alex-NV56R:/usr/local/tomcat/apache-tomcat-8.0.26/webapps#
```

2.-Nuestro form1.jsp estará estructurado así, donde tenemos un formulario para capturar los datos y después enviarlo al Controlador 1

```
form1.jsp
/usr/local/tomcat/apache-tomcat-8.0.26/webapps/Sitio3

<HTML>
<body background="images/pergamino.jpg">
<CENTER>

    <H1>Calcular el area y volumen de un cilindro</H1>

    <FORM ACTION="./Controlador1" METHOD="POST">

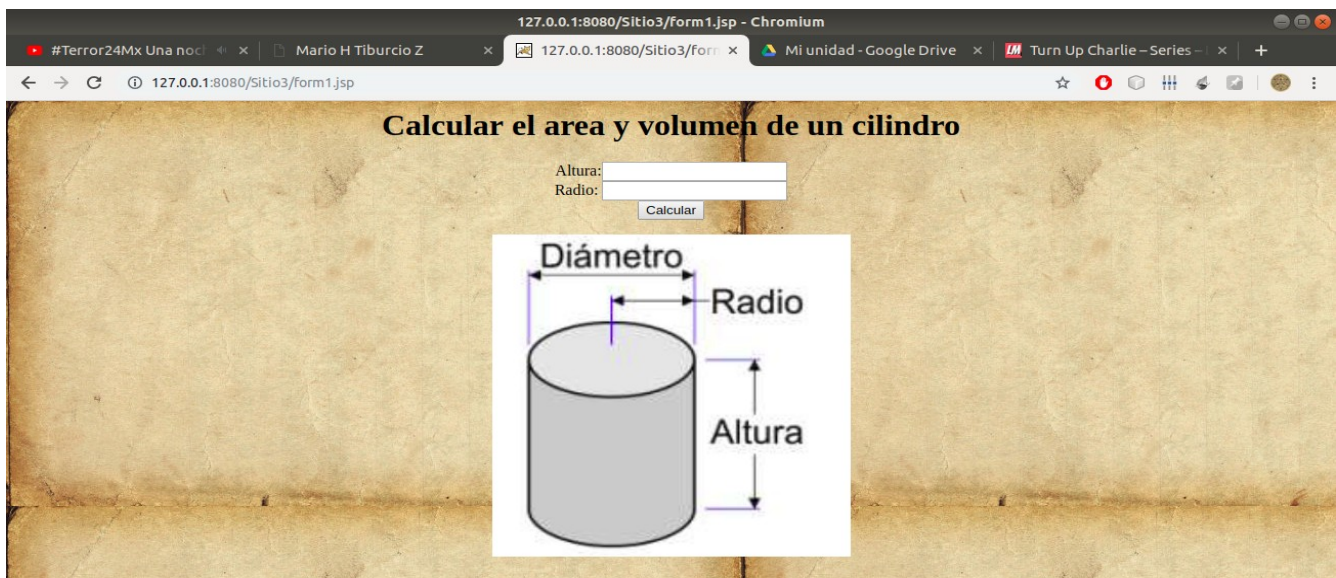
        Altura:<INPUT TYPE="TEXT" NAME="altura" title = "Cifras aceptadas cantidad . cantidad o .cantida" required pattern = "([^\.][0-9]*.[0-9]*)|([^\.][0-9]+)|([0-9])"><BR>

        Radio:      <INPUT TYPE="TEXT" NAME="radio" title = "Cifras aceptadas cantidad . cantidad o .cantida" required pattern = "([^\.][0-9]*.[0-9]*)|([^\.][0-9]+)|([0-9])"><BR>

        <INPUT TYPE="SUBMIT" VALUE="Calcular">

    </FORM>
    
</CENTER>

</HTML>
```



3.-Nuestro controlador1.java recibirá la información que enviamos del formulario con el fin de hacer las operaciones correspondientes, lo compilamos y lo mandamos a la carpeta classes

```
Controlador1.java
/usr/local/tomcat/apache-tomcat-8.0.26/webapps/Sitio3/WEB-INF/src

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
import geometria.cuerpos.Cilindro;

public class Controlador1 extends HttpServlet
{

    public void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {

        String radio=request.getParameter("radio");
        double radioD=Double.parseDouble(radio);
        String altura=request.getParameter("altura");
        double alturaD=Double.parseDouble(altura);
        Cilindro ci=new Cilindro();
            ci.setRadio(radioD);
            ci.setAltura(alturaD);
            double area3=ci.area();
            double vol=ci.volumen();
        request.setAttribute("RADIO",radioD);
        request.setAttribute("ALTURA",alturaD);
        request.setAttribute("AREA",area3);
        request.setAttribute("VOLUMEN",vol);
        RequestDispatcher vista=request.getRequestDispatcher("./Salida.jsp");
        vista.forward(request,response);

    }

}
```


4.-Creamos un archivo que se llame web.xml justo en la carpeta WEB-INF, este archivo nos hará un mapeo y nos permitirá visualizar hacia donde dirigirnos para encontrar el servlet, con que nombre y la dirección que visualizaremos

```
web.xml
/usr/local/tomcat/apache-tomcat-8.0.26/webapps/Sitio3/WEB-INF

<!-- ?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml$
version="2.4"> -->
<web-app>
<display-name>
Ejemplos de un Servlet
</display-name>
<description>
Ejemplo para prueba
</description>
<!-- Publicamos nuestro servlet aquí -->
<servlet>
<servlet-name>UnEjemploDeServlet</servlet-name>
<servlet-class>Controlador1</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UnEjemploDeServlet</servlet-name>
<url-pattern>/Controlador1</url-pattern>
</servlet-mapping>
</web-app>
```

5.-Otro jsp nos funcionara como receptor para nuestro servlet nos mostrara los datos y la forma en la que estos serán visualizados

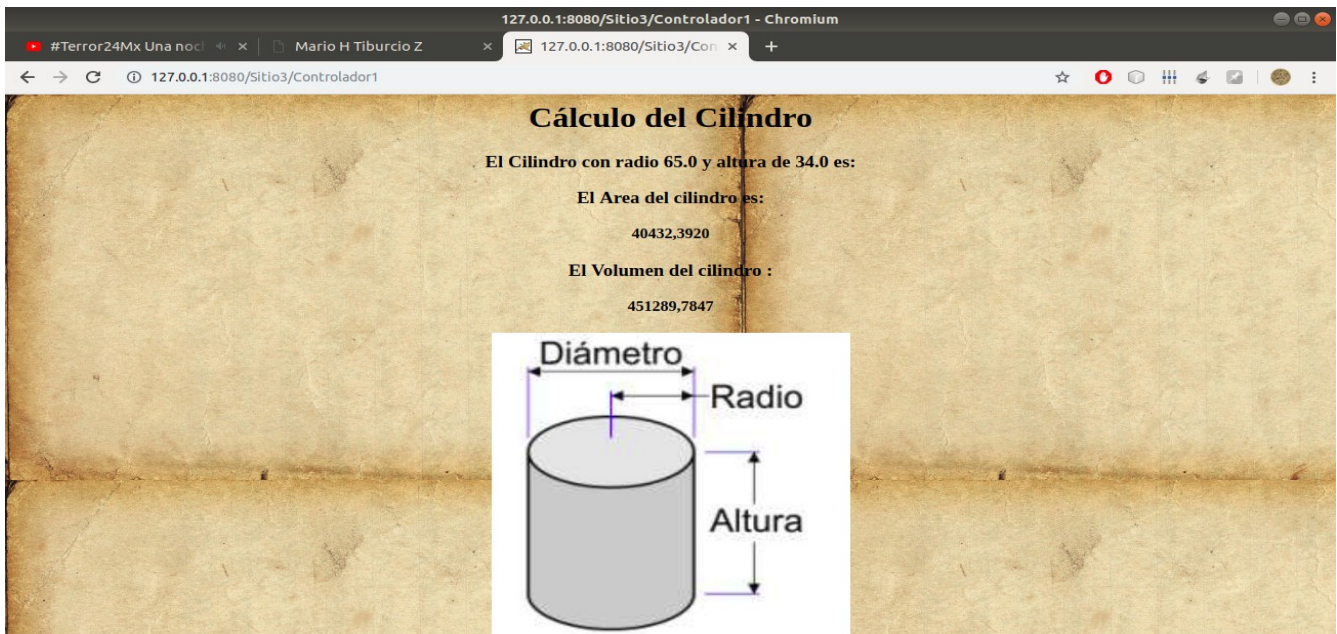
```
Salida.jsp
/usr/local/tomcat/apache-tomcat-8.0.26/webapps/Sitio3

<html>
<body background="images/pergamino.jpg">
  <center>

    <h1>Cálculo del Cilindro</h1>

  <%
    double radio=(double)request.getAttribute("RADIO");
    double altura=(double)request.getAttribute("ALTURA");
    double area=(double)request.getAttribute("AREA");
    double vol=(double)request.getAttribute("VOLUMEN");
  %>

  <h3>El Cilindro con radio <%=radio%> y altura de <%=altura%> es:</h3>
  <h3>El Area del cilindro es:</h3>
  <h4><%=String.format("%10.4f",area)%></h4>
  <h3>El Volumen del cilindro :</h3>
  <h4><%=String.format("%10.4f",vol)%></h4>
  
  </center>
</html>
```



6.-Tenemos que incluir en la carpeta lib dentro de la carpeta WEB-INF un archivo jar con clases previamente compiladas que incluyen los beans que usaremos en el controlador1

```
root@alex-NV56R:/usr/local/tomcat/apache-tomcat-8.0.26/webapps/Sitio3/WEB-INF/li
b# ls
alexapi.jar
root@alex-NV56R:/usr/local/tomcat/apache-tomcat-8.0.26/webapps/Sitio3/WEB-INF/li
b#
```

Conclusión

Para montar un sitio sobre tomcat tiene que tener cierta estructura y si se trata de un modelo vista controlador, debe contener cierta estructura y elementos que hagan posible la vinculación y posterior uso de esto, en este caso lo hicimos con el fin de calcular el área y volumen de un cilindro y con una librería previamente hecha y cargada como un jar