

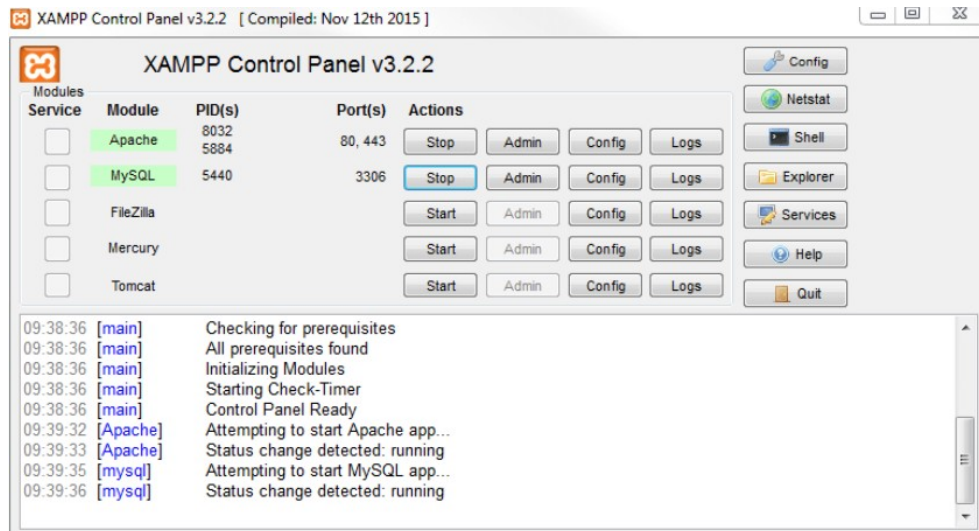
Actividades evaluables.

Documentar y publicar las siguientes actividades evaluables.

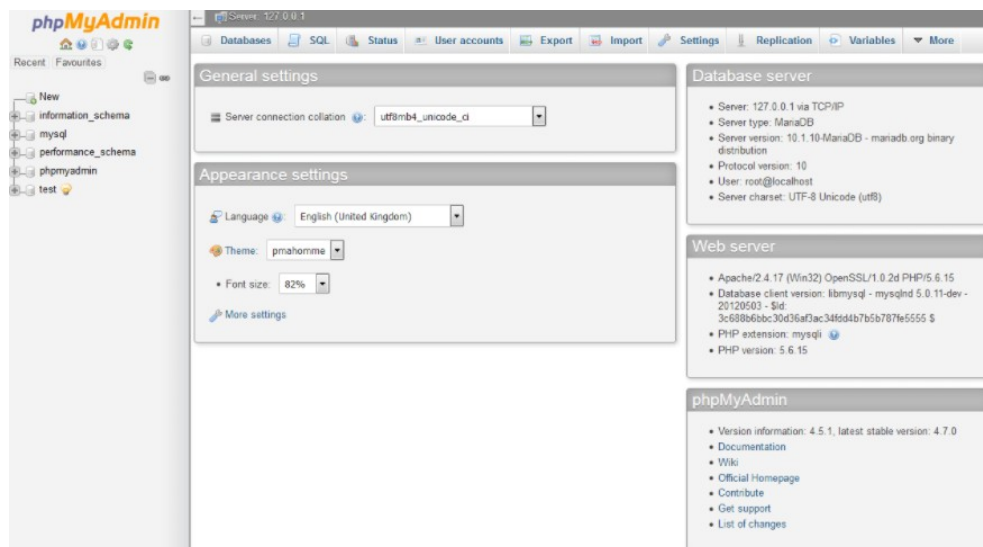
Actividad 1.

Objetivo: Instalar y configurar un paquete software que integre Apache, PHP y MSQL.

1. Instalar y configurar XAMPP.



2. Instalar phpmyadmin.



3. Probar la aplicación creando un porfolio personal básico.

4. Documentar y publicar la práctica utilizando pantallazos personalizados del desarrollo.

Actividad 2.

Objetivo: Instalación y configuración de un entorno web utilizando una máquina virtual.

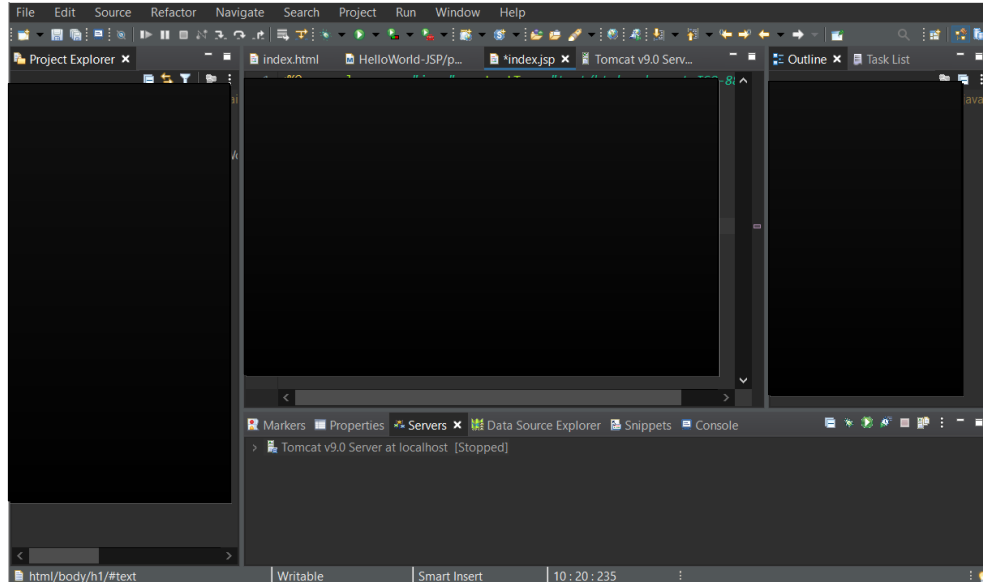
1. Descargar e instalar VirtualBox.
2. Crear una máquina virtual con Ubuntu Server e instalar de manera independiente Apache, MariaDB, PHP y phpMyAdmin.
3. Configurar la conexión entre el entorno de trabajo y el servidor.
4. Probar la aplicación creando un portfolio personal básico.
5. Documentar y publicar la práctica utilizando pantallazos personalizados del desarrollo.

Actividad 3.

Objetivo: Crear una aplicación que permita calcular el doble de un número utilizando como tecnologías del lado del servidor, Java, JSP, Servlets y Apache TomCat.

1. Instalar y configurar el entorno de trabajo.

Para el desarrollo de la actividad se aconseja utilizar como entorno de desarrollo integrado (IDE) Eclipse. Existen alternativas interesantes como pueden ser NetBeans o IntelliJ IDEA.

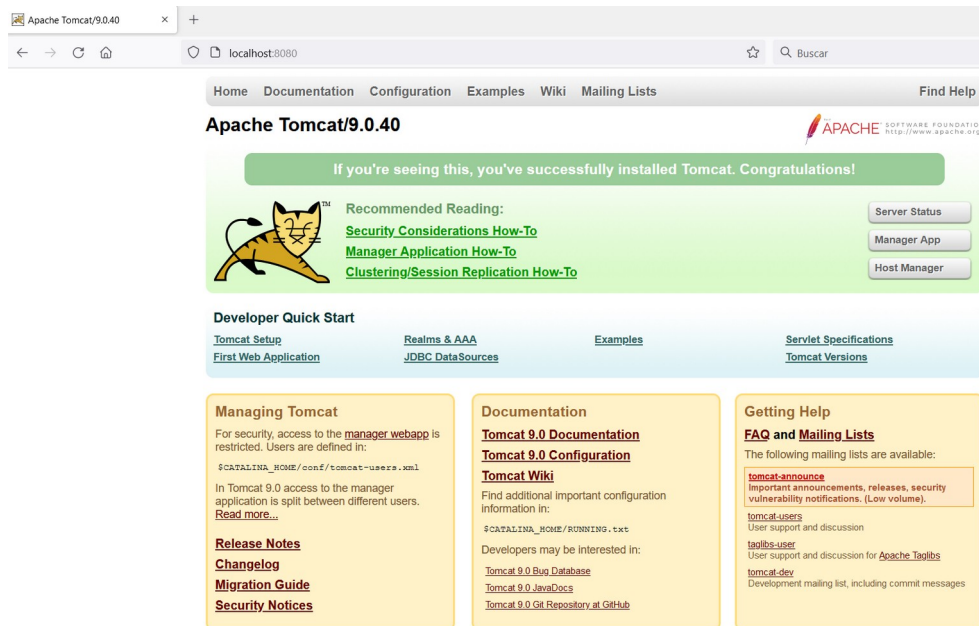


Recursos:

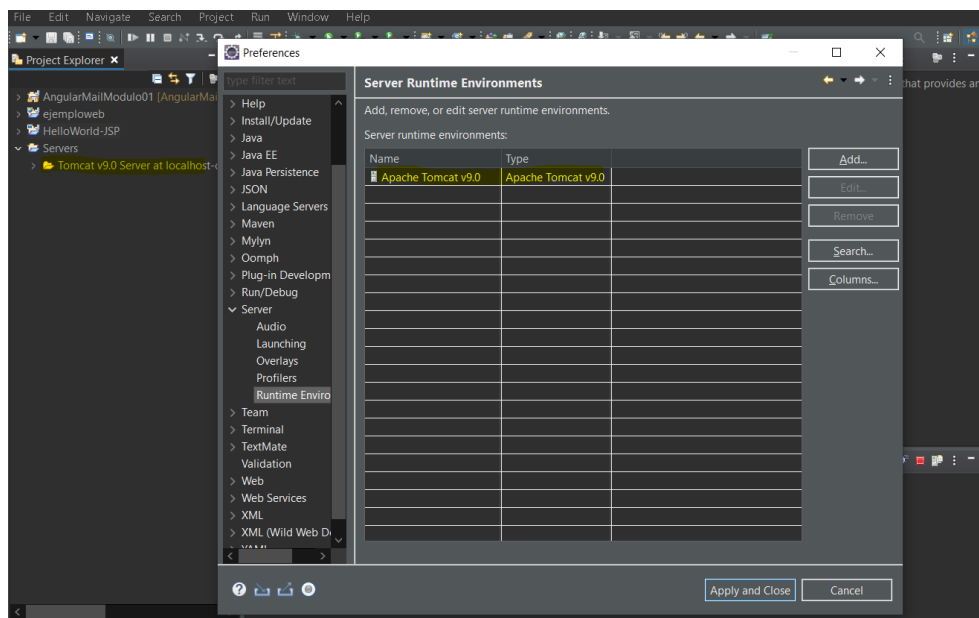
[Eclipse](#)

2. Instalar y configurar el servidor de servlets.

Apache Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages de Oracle.



La instalación en el equipo se puede realizar de manera manual descargando e instalando el software, o desde el propio entorno de desarrollo. Eclipse integra perfectamente el manejo del servidor, facilitando las fases de desarrollo y despliegue. Se aconseja la integración del servidor en Eclipse.

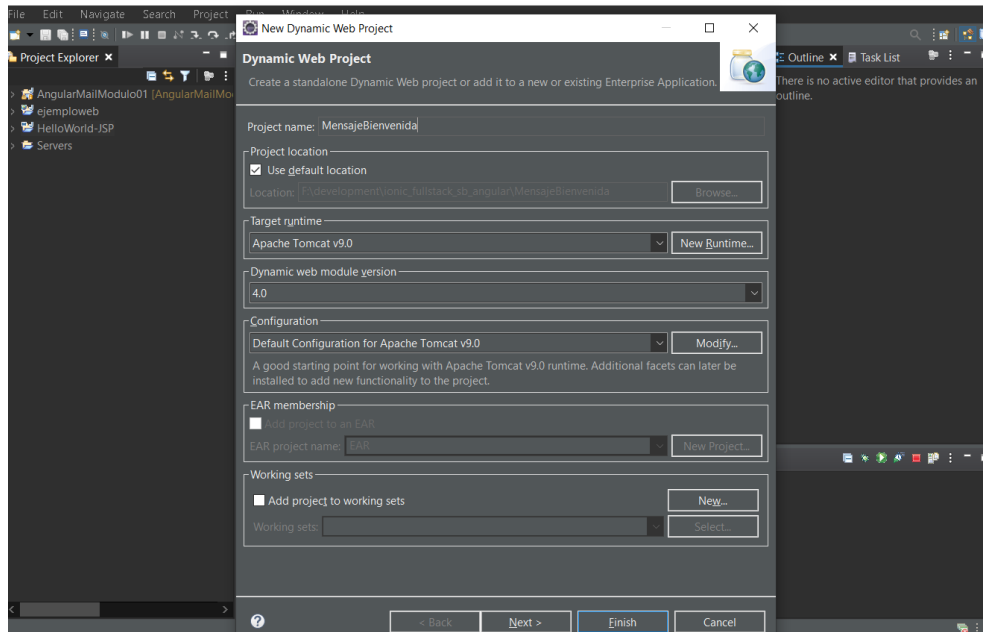


Recursos:

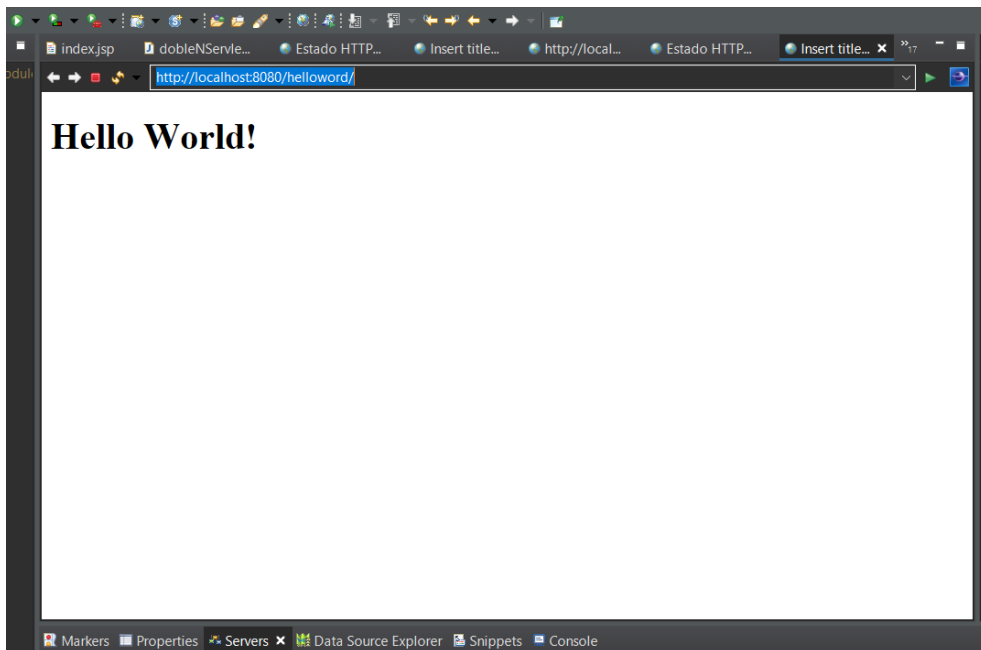
[Tomcat](#)
[Integración de Tomcat en Eclipse](#)

3. Desarrollar una página *JSP* que muestre el mensaje “Hello World”.

- Crear un nuevo proyecto del tipo Dynamic Web Project. Utilizar los valores establecidos por defecto.



- Crear un nuevo fichero JSP: index.jsp
- Modificar el fichero index.jsp añadiendo un encabezado con “Hello Word”
- Probar la aplicación y compraba que funciona correctamente.



4. Documentar y publicar la práctica utilizando pantallazos personalizados del desarrollo.

5. Desarrollar una aplicación que calcule el doble de un número.

- Modificar el fichero index.jsp para que aparezca como encabezado “Cálculo del doble de un número”.
- Si es necesario, convertir el proyecto a Maven para automatizar la gestión de las dependencias.
- Añadir al fichero index.jsp un formulario que permita introducir al usuario un número.

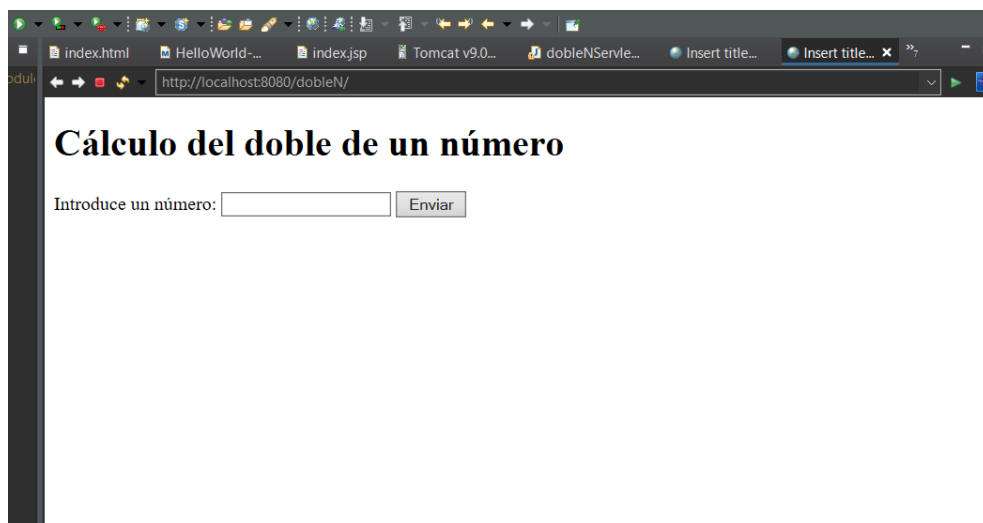
```
<form action="dobleNServlet" method="post">
  Introduce un número:
  <input type="text" name="numero" size="20">
  <input type="submit" value="Enviar" />
</form>
```

- ¿Para qué se utiliza el atributo action?
- ¿Para qué se utiliza el atributo method?
- ¿Para qué se utiliza el atributo name?
- Crear un nuevo fichero de tipo Servlet, que contendrá la clase que dará respuesta al formulario. Utiliza como nombre el valor del atributo action en el formulario. En el ejemplo “dobleNServlet”.

```
// TODO Auto-generated method stub
String strNumero = request.getParameter("/*?????*/");
int numero;
int doble;
numero = /*?????*/;
doble = /*?????*/;
PrintWriter writer = response.getWriter();
writer.println("<h1>DOBLE: " + /*?????*/ + "</h1>");
writer.close();
```

- Modificar el método doPost que calculará el doble del número introducido en el formulario. Sustituye los caracteres ‘?’ por los valores correctos.

6. Probar la aplicación para comprobar el correcto funcionamiento.



7. Documentar y publicar la práctica utilizando pantallazos personalizados del desarrollo.

Actividad 4.

Objetivo: Crear un entorno virtual para desarrollo en el lado del servidor.

Recursos:

[VirtualBox](#)

[Vagrant](#)

[Git](#)

[Laravel Homstead](#)

Observaciones:

- Es necesario que exista el archivo `id_rsa` referenciado en el archivo `Homestead.yaml`. Si no existe debe crearse en el directorio `.ssh`.

```
| ssh-keygen -t rsa -C "correo@correo.es"
```

- Podemos crear también el fichero vacío.
- En ocasiones habrá que deshabilitar y habilitar la interfaz de red en el administrador de dispositivos.

1. Descargar e instalar Vagrant.
2. Comprobar que la instalación se realizó correctamente lanzando el comando que muestra la versión instalada.
3. Añadir el proyecto Laravel Homstead seleccionando VirtualBox:

```
F:\development\HashiCorp\Vagrant>vagrant box add laravel/homestead
==> box: Loading metadata for box 'laravel/homestead'
    box: URL: https://vagrantcloud.com/laravel/homestead
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) hyperv
2) parallels
3) virtualbox

Enter your choice: 3
```

4. Clonar el proyecto de Laravel Homstead en el directorio de trabajo:

```
| git clone https://github.com/laravel/homestead.git
```

5. Inicia homstead

```
| init.bat
```

6. Observar el archivo de configuración Homestead.yaml, especialmente en folders, y sites. Es posible tener múltiples proyectos en la misma máquina.

```
ip: "192.168.10.10"
memory: 2048
cpus: 2
provider: virtualbox

authorize: ~/.ssh/id_rsa.pub

keys:
  - ~/.ssh/id_rsa

folders:
  - map: f:\development\Homestead\code
    to: /home/vagrant/code

sites:
  - map: homestead.test
    to: /home/vagrant/code/public

databases:
  - homestead

features:
  - mysql: true
  - mariadb: false
  - postgresql: false
  - ohmyzsh: false
  - webdriver: false

#services:
```

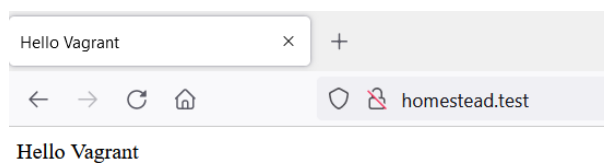
- ## 7. Lanza la máquina.

vagrant up

- ## 8. Prueba a conectarte a la máquina por ssh.

[illegible]

- Modifica el fichero de hosts para asignar nombre y dirección ip del archivo de configuración Homestead.yaml.
- Crea una página que muestre el mensaje "Hello Vagrant".



11. Documentar y publicar la práctica utilizando pantallazos personalizados del desarrollo.

Actividad 5.

Objetivo: Utilizar entornos de desarrollos avanzados.

1. Instalar y configurar una pila LAMP utilizando contenedores docker.
2. Instalar y configurar una pila LAMP utilizando servicios AWS.
3. Documentar y publicar las prácticas utilizando pantallazos personalizados del desarrollo.