



Práctica 1 – Implantación de arquitecturas web.

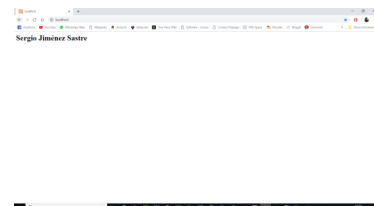
El principal objetivo de esta práctica es realizar la instalación de los servidores web más utilizados Apache y Tomcat, y comprobar que funciona correctamente. Antes de comenzar, resaltar que cada alumno puede realizar la instalación tanto en Windows como en Ubuntu (o ambos), y puede hacerlo en una máquina virtual si lo considera oportuno. Cada uno de los siguientes pasos deberían estar documentados, es decir, se debe realizar las capturas de pantalla a modo de tutorial y comentar cada una de ellas, además de resaltar las conclusiones obtenidas. También, se valorará el hecho de realizar comprobaciones más allá de las solicitadas en la práctica.

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

1. Realiza la instalación de XAMPP. Comprueba el correcto funcionamiento modificando la página.html por defecto, de manera que en ésta aparezca tu nombre y apellidos. Al instalar el paquete de XAMPP se crea una estructura dentro de la ruta que le des al instalarlo (por defecto en <C://xampp>). Dentro de esta carpeta hay una carpeta llamada htdocs, que es donde se encuentra nuestra página index.php (con extensión .php, no .html).

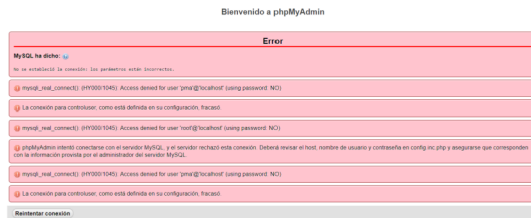
Para modificar nuestro index.php por defecto para que salga nuestro nombre y apellidos insertamos el siguiente código en el archivo. Para arrancarlo clickamos en start en Apache desde el Xampp Control Panel, abrimos un navegador e introducimos <http://localhost> en la barra de direcciones.

```
<?php
    echo '<h1>Sergio Jiménez Sastre</h1>'
?>
```

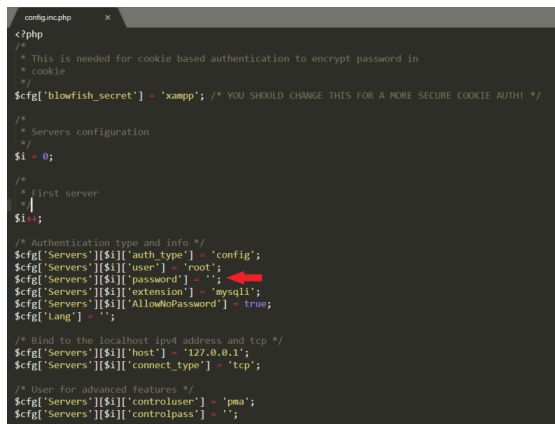


2. Comprueba que tu servidor está disponible desde el equipo de un compañero y es accesible la página. Para ello mi compañero introdujo mi dirección IP en la barra de direcciones del navegador, en mi caso he de tener el servicio de Apache arrancado. ¿Es accesible desde fuera del instituto (utiliza tu móvil por ejemplo)? No es accesible. ¿Por qué? Ya que no se encuentra en la misma red en la que está mi ordenador local.
3. Crea una página .php que imprima el texto “nombre y apellidos + php funciona”, y comprueba que apache sirve tu página php. Como se a nombrado en el ejercicio número 1 el archivo no interpretará el código php si el archivo está con la extensión.html.
4. Comprueba que está accesible MySQL a través de la herramienta PHPMyAdmin. Al principio no nos deja acceder al phpMyAdmin. En el panel de control de XAMPP clickaremos a admin para acceder (en el apartado de MySQL y arrancándolo).

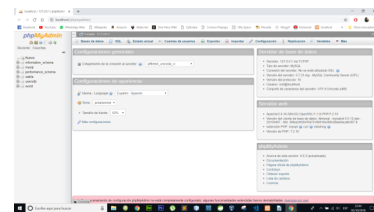
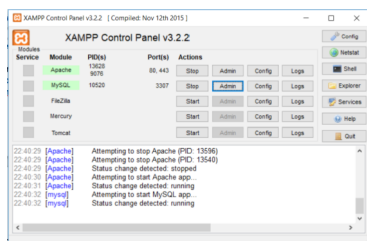
Al principio no nos deja acceder al phpMyAdmin



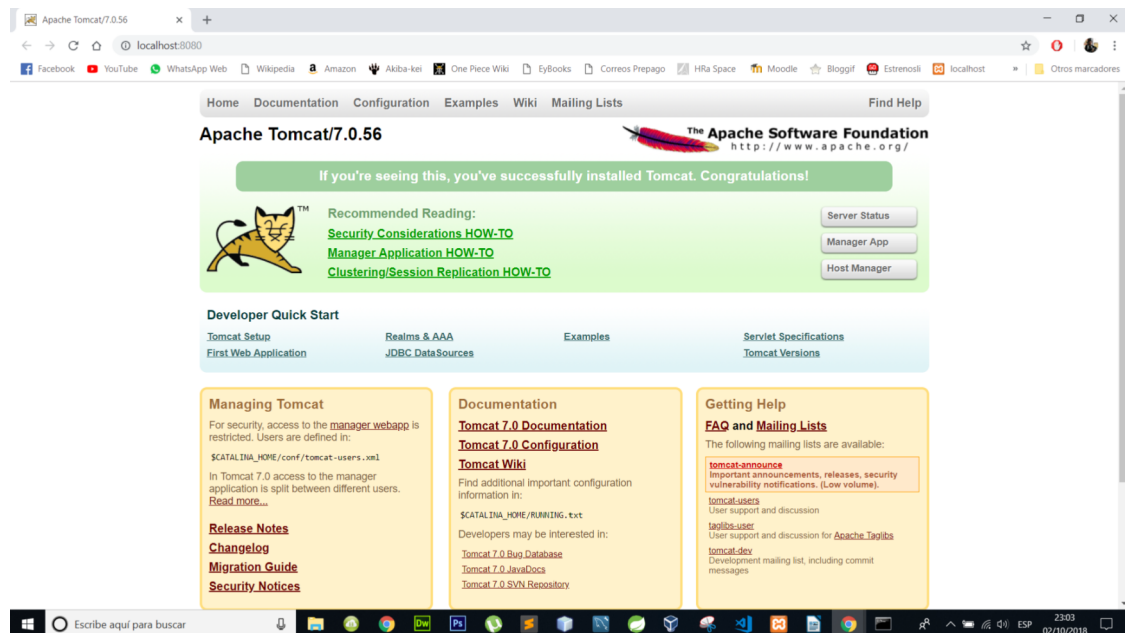
Tendremos que ir al un archivo en la carpeta phpMyAdmin, allí ir al archivo config.inc.php y añadirle la contraseña del MySQL.



Una vez realizados los cambios arrancaremos Apache y MySQL. Una vez iniciados para acceder tendremos que darle en el panel de control de xampp en el botón de admin o bien dirigirnos a <http://localhost/phpmyadmin> (por defecto trabaja e el puerto 3306, en mi caso se cambió al 3307 ya que éste estaba ocupado).

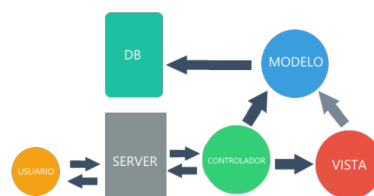


5. Arranca Tomcat (XAMPP incorpora también Tomcat) y comprueba que es accesible: <http://localhost:8080/>. Desde esta dirección podremos arrancar el tomcat arrancando el servicio primero desde el panel de control de xampp.



6. ¿Puedes acceder al manager app o server status? Si no es así, averigua el motivo y modifica la configuración para que esta sea accesible. Revisa el fichero Apache Software Foundation\Tomcat 9.0\conf\tomcat-users.xml y crea un nuevo usuario con el rol manager-gui y comprueba que es accesible.
7. Crea una página .jsp que imprima el texto “nombre y apellidos + JSP funciona”, y comprueba que apache sirve tu página jsp.
8. En las sesiones de teoría se ha hablado del modelo cliente servidor, concretamente del modelo multicapa. Investiga, y comenta que diferencia existe entre multicapa y multinivel.
9. ¿Qué es el modelo MVC (modelo-vista-controlador)?

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (el modelo, la vista y el controlador).



¿Qué tarea deben realizar desempeñan cada uno de estos componentes?

El Modelo: Contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

El modelo es el responsable de:

Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.

Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del

proveedor".
Lleva un registro de las vistas y controladores del sistema.
Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

La Vista: Compone la información que se envía al cliente y los mecanismos interacción con éste.

Las vistas son responsables de:
Recibir datos del modelo y los muestra al usuario.
Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

El Controlador: Actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El controlador es el responsable de:
Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
Contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega (nueva_orden_de_venta)".

¿Qué diferencias encuentras entre el modelo MVC y modelo multicapa? Por un lado, MVC es un patrón arquitectural; define en qué bloques (o capas) estructuramos lógicamente nuestra aplicación (Modelo, Vista y Controlador), pero además detalla las responsabilidades exactas de cada capa y la forma que tienen de relacionarse entre sí.

Por tanto, si programas ciñéndote al patrón MVC estarás dividiendo tu sistema en tres capas, pero no al contrario: puedes programar en 3 capas sin necesidad de seguir dicho patrón.

Por otro lado, la división tradicional en 3 capas (presentación, lógica de negocio y datos) no es la misma que propone MVC. La capa Modelo incluiría la lógica de negocio y datos, la Vista incluiría exclusivamente el interfaz externo (una parte de la presentación), y el papel de la capa Controlador no tendría una clara correspondencia en la estructura clásica (quizás una mezcla de presentación y lógica de negocio).