

Ejercicio 1: LAMP y WISA. Explica en qué consiste cada una de ellas.

**LAMP:** funciona bajo un esquema de código abierto, es de bajo costo. Es una plataforma bastante rápida y ofrece buen rendimiento sobre todo a aplicaciones web que no sean demasiado grandes o que tenga excesiva actividad. Además existe para ella una amplia gama de aplicaciones, perfectamente adaptable a las necesidades de pequeñas y medianas empresas o de aquellas que se encuentren en los primeros estadios de la gestión de información, que incluyen blogs, manejadores de contenidos, aplicaciones para trabajo en grupo, entre otras. Algunas de estas aplicaciones son pagas pero la mayoría son gratuitas.

**WISA:** es más costosa y provee mayor nivel de soporte. En general es más robusta si se tiene en mente aplicaciones web que funcionen en una intranet o que tengan un gran nivel de actividad (varios miles de transacciones electrónicas al día, por ejemplo). También existe abundancia de aplicaciones web desarrolladas por terceros, aunque la mayoría de ellas son pagas.

**Otras plataformas:**

MAMP, APACHE, MySQL, PHP.

Ejercicio 2: La escalabilidad horizontal y cluster.

**La escalabilidad horizontal** consiste en potenciar el rendimiento del sistema desde un aspecto de mejora global, a diferencia de aumentar la potencia de una única parte del mismo. Este tipo de escalabilidad se basa en la modularidad de su funcionalidad. Por ello suele estar conformado por una agrupación de equipos que dan soporte a la funcionalidad completa. Normalmente, en una escalabilidad horizontal se añaden equipos para dar mas potencia a la red de trabajo.

Si se aplica un modelo de escalabilidad basado en la horizontalidad, no existen limitaciones de crecimiento a priori. Como principal e importante defecto, este modelo de escalabilidad supone una gran modificación en el diseño, lo que conlleva a una gran trabajo de diseño y reimplantación. Si la lógica se ha concebido para un único servidor, es probable que se tenga que estructurar el modelo arquitectónico para soportar este modelo de escalabilidad.

**cluster:** es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único equipo, más potente . Con ello se pretende mejorar los siguientes parámetros de la arquitectura:

- Alto rendimiento
- Alta disponibilidad
- Equilibrio de carga
- Escalabilidad

Ejercicio 3: arquitectura Web de tres capas, ¿cuáles son y cuál es la función de cada una de ellas?

**Capa 1:** Cliente de aplicación: Navegador Web

**Capa 2 :** Servidor de Aplicaciones: Apache, Servidor Tomcat con servlet's

**Capa 3 :** Servidor de Datos: base de datos, servidor SMTP...

El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de presentar los datos un formato adecuado.

El segundo nivel está referido habitualmente a algún tipo de programa o script.

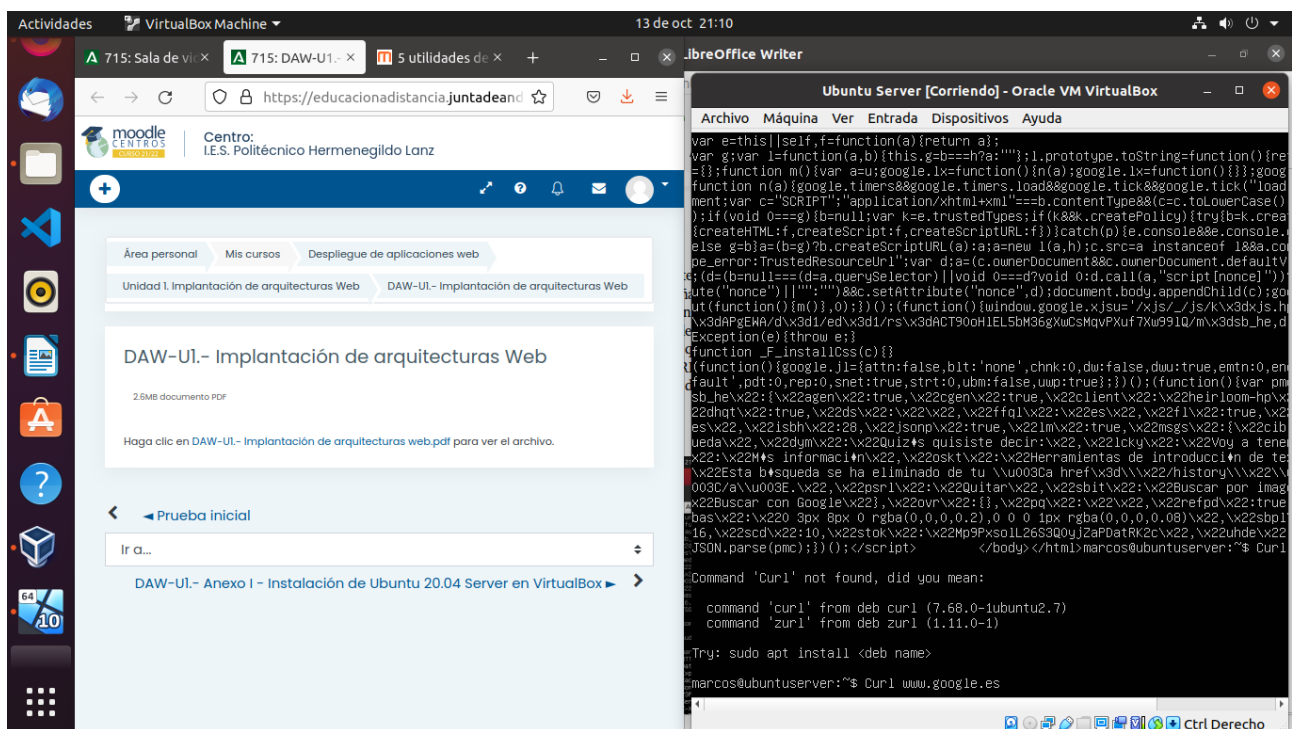
Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución. Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

## Ejercicio 5:

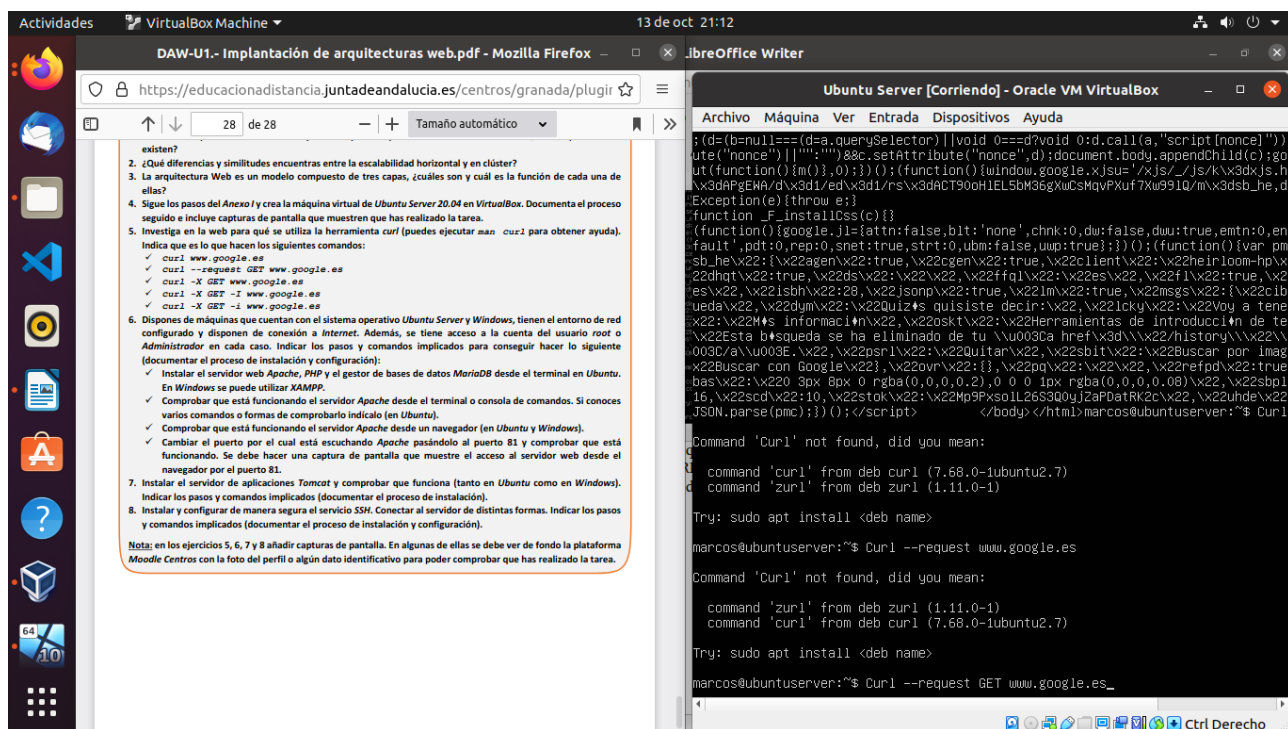
### Comando Curl:

Es un comando disponible en la mayoría de los sistemas basados en Unix. Es una abreviatura de «Client URL». Los comandos de Curl están diseñados para funcionar como una forma de verificar la conectividad a las URL y como una gran herramienta para transferir datos.

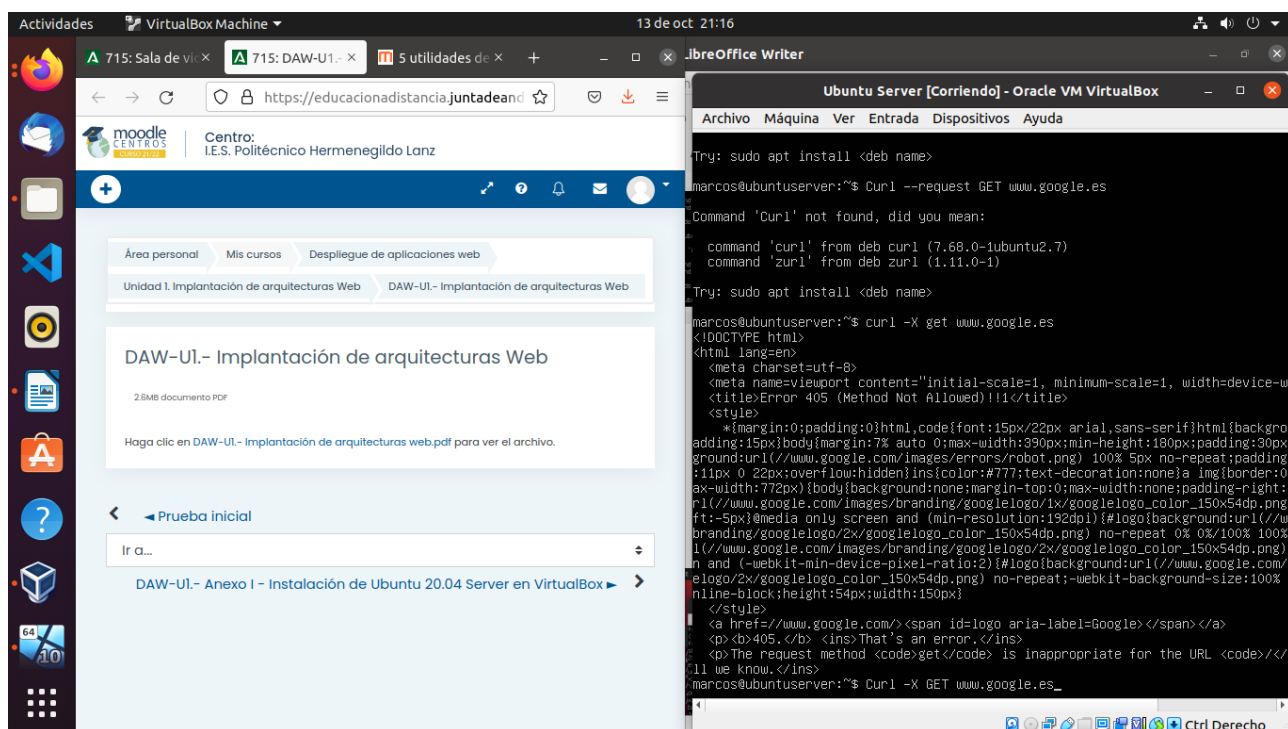
**Curl** [www.google.com](http://www.google.com) : esto te pega el contenido de la página web ingresada.



**Curl --request GET** [www.google.es](http://www.google.es) : Le dice a curl que use un "destino" alternativo (ruta) en lugar de usar la ruta como se proporciona en la URL. Particularmente útil cuando se desea emitir solicitudes HTTP sin barra inclinada u otros datos que no siguen el patrón de URL normal, como "OPCIONES".



**Curl -X GET [www.google.es](http://www.google.es)** : La solicitud especificada se utilizará en lugar del método utilizado de otro modo (que por defecto es GET)



**Curl -X GET -I [www.google.es](http://www.google.es)** : Se mostrarán los encabezados de todas las páginas solicitadas

