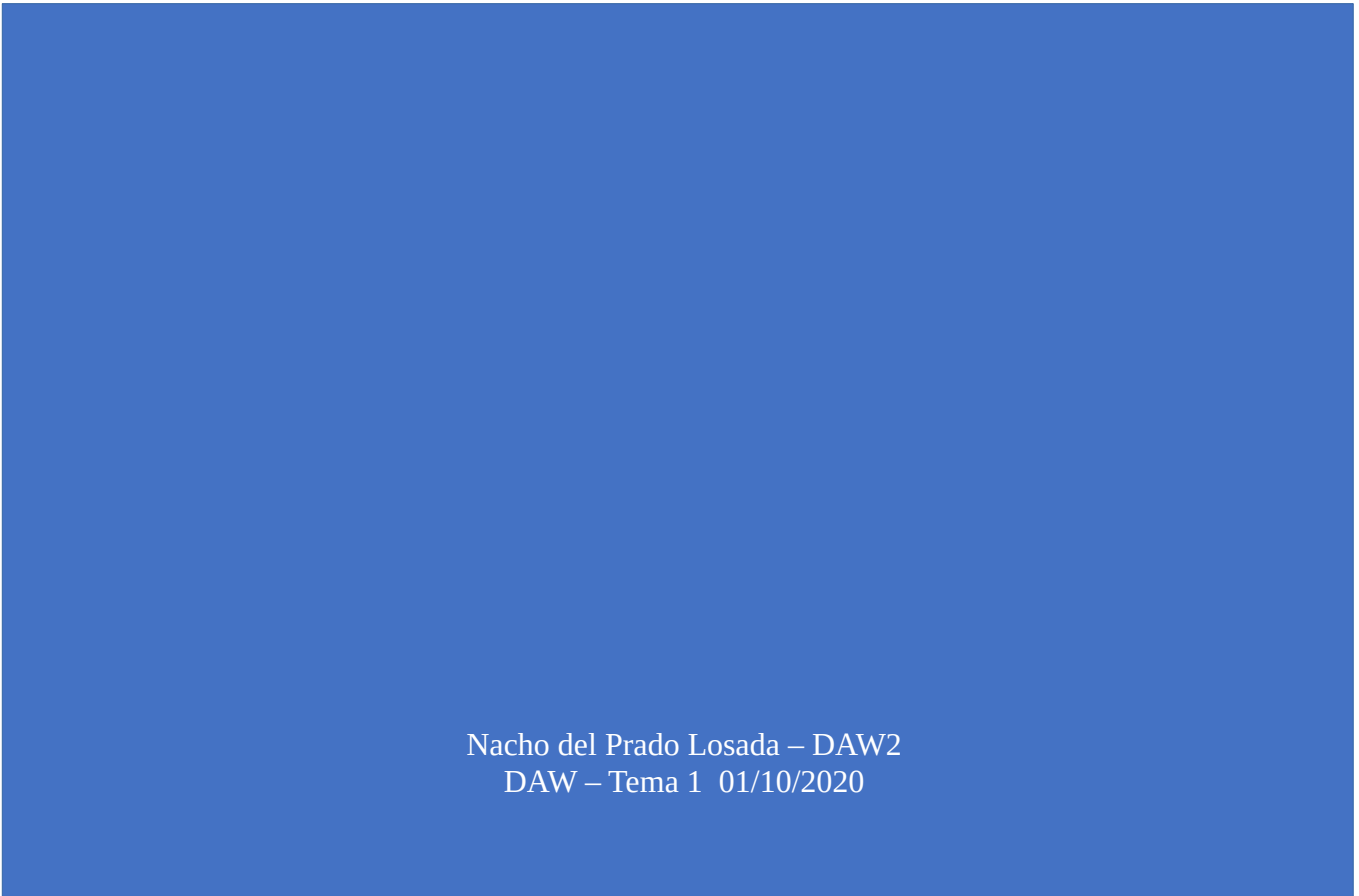




## TAREA 2



Nacho del Prado Losada – DAW2  
DAW – Tema 1 01/10/2020

## Sumario

1. Especificaciones técnicas de tu ordenador personal.....	3
2. Estudio detallado del protocolo HTTP.....	4
3. Se pide buscar en Internet diferentes aplicaciones web y clasificarlas según su tipo, como mínimo dos páginas de cada tipo.....	5
4. ¿Qué tecnologías se utilizan en cada una de las capas de la arquitectura web?.....	6
5. Existen multitud de servidores web en el mercado. La empresa Netcraft se encarga de hacer estadísticas y mediciones sobre la utilización de los distintos servidores web. Se pide una gráfica con la utilización de los principales servidores web y las principales características de los cinco primeros.....	7
6. Diferencias y semejanzas entre Saas, PaaS e IaaS.....	8
7. Diferentes versiones disponibles de Apache Tomcat y sus diferentes versiones de las especificaciones.....	9
8. URI, URN Y URL: definición, sintaxis (estructura) y diferentes ejemplos.....	11
9. Escalabilidad horizontal: Ventajas y inconvenientes.....	12
10. Escalabilidad vertical: Ventajas y inconvenientes.....	13
Bibliografía.....	14

## 1. Especificaciones técnicas de tu ordenador personal

ESPECIFICACIONES	
Sistema operativo	Windows 10 Home x64
Memoria RAM	8 GB
Particiones	C: 915 GB D: 14,4 GB
Nombre del equipo	LAPTOP-00RLUILC
Procesador	Intel® Core™ i3-6006U CPU @ 2.00GHz 2.00 GHz

## 2. Estudio detallado del protocolo HTTP.

HTTP o "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos **HTML**. Es la base de cualquier intercambio de datos en la web, y protocola de **estructura cliente-servidor**.

Clientes y servidores se comunican intercambiando mensajes individuales (en contraposición a las comunicaciones que utilizan flujos continuos de datos). Los mensajes que envía el cliente, normalmente un navegador Web, se llaman *peticiones*, y los mensajes enviados por el servidor se llaman *respuestas*.

- Cliente: esta función es realizada en la mayor parte de los casos por un navegador Web. El navegador es siempre el que inicia una comunicación (petición). El navegador envía una petición de documento HTML al servidor. Entonces procesa este documento, y envía más peticiones para solicitar scripts, hojas de estilo (CSS), y otros datos que necesite (normalmente vídeos y/o imágenes). El navegador, une todos estos documentos y datos, y compone el resultado final: la página Web. Los scripts, los ejecuta también el navegador, y también pueden generar más peticiones de datos en el tiempo, y el navegador, gestionará y actualizará la página Web en consecuencia.
- Servidor web: "sirve" los datos que ha pedido el cliente. Un servidor no tiene que ser necesariamente un único equipo físico, aunque si que varios servidores pueden estar funcionando en un único computador. Está formado por varios programas como cache, bases de datos, servidores de correo electrónico, etc, y que generan parte o todo el documento que ha sido pedido.

HTTP esta pensado y desarrollado para ser leído y fácilmente interpretado por las personas, haciendo de esta manera más facil la depuración de errores, y reduciendo la curva de aprendizaje para las personan que empieza a trabajar con él.

## Tarea 2

Este protocolo es fácil de ampliar. Funcionalidades nuevas pueden desarrollarse, sin más que un cliente y su servidor.

## Tarea 2

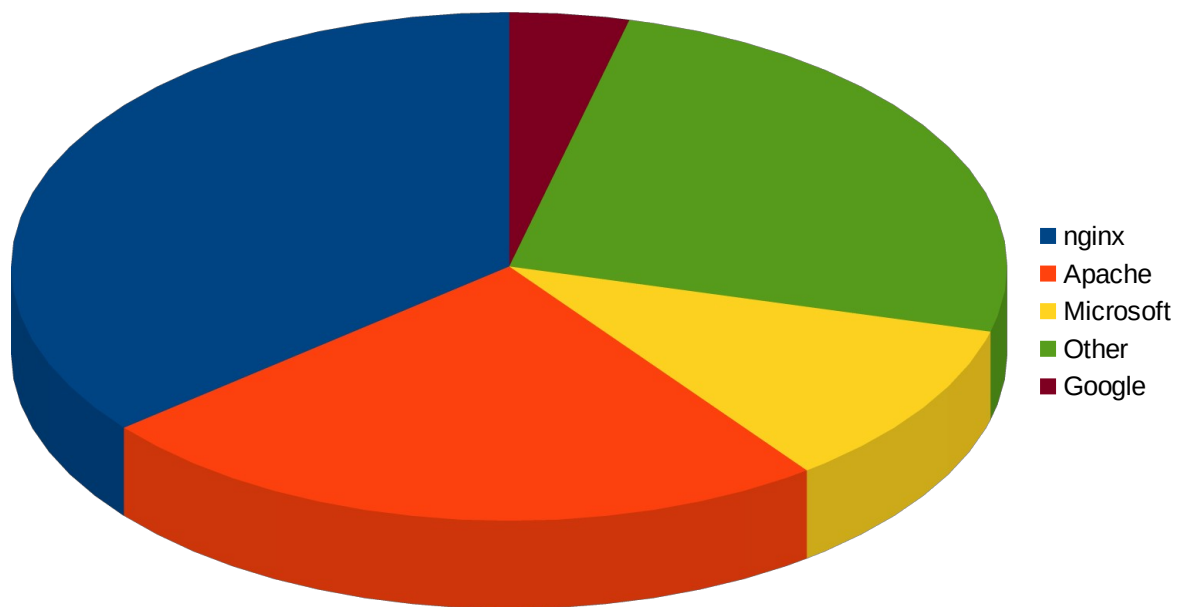
**3. Se pide buscar en Internet diferentes aplicaciones web y clasificarlas según su tipo, como mínimo dos páginas de cada tipo.**

TIPOS	APLICACIONES	
Estática	Cosmoeduca	Junta de Andalucía
Dinámica	Google	Brainly
Portal	Wikipedia	Amazon
Animada	Android	Lander
E-commerce	Walmart	Massimo Dutti
Gestor de contenidos	WordPress	Joomla!

**4.¿Qué tecnologías se utilizan en cada una de las capas de la arquitectura web?**

- Capa de presentación: HTML, CSS, JavaScript.
- Capa de negocio: Servlets, JSP.
- Capa de datos: PHP, Python, Rails, Go, C#, Java, NodeJS, MongoDB, MySQL

**5.Existen multitud de servidores web en el mercado. La empresa Netcraft se encarga de hacer estadísticas y mediciones sobre la utilización de los distintos servidores web. Se pide una gráfica con la utilización de los principales servidores web y las principales características de los cinco primeros.**



- Nginx: de código abierto, pero con una versión comercial llamada Nginx Plus. Es multiplataforma, está programado en C y es muy usado por sus capacidades.
- Apache: de código abierto, multiplataforma, programado en C y con una amplia aceptación en la red, a pesar de no ser tan rápido como Nginx, por ejemplo.
- Microsoft IIS: software propietario, funciona en Windows, está programado en C++.



## 6.Diferencias y semejanzas entre Saas, PaaS e IaaS

- **IaaS:** *Infrastructure as a Service*. Sistema idóneo para desarrolladores que deseen encargarse de la gestión y administración de su infraestructura. Ofrece un mayor control que otras alternativas como **PaaS**, de modo que el desarrollador es el responsable de todo lo relacionado con el mantenimiento de la infraestructura, incluso de escalar sus aplicaciones en función de cuáles sean sus necesidades. Ej: Amazon Web Service.
- **Paas:** *Platform as a Service*. Alternativa para aquellos desarrolladores de aplicaciones que únicamente quieren preocuparse de construir la app. La infraestructura la proporciona la plataforma y se ocupa tanto de su gestión como de su mantenimiento. Ej: Jelastic.
- **Saas:** *Software as a Service*. Cualquier servicio que esté basado en la web. En este caso los usuarios acceden al servicio sin prestar la más mínima atención al software. Tanto el desarrollo como el mantenimiento y resto de gestiones son responsabilidad única del proveedor. Así, los usuarios tienen un control mínimo sobre el servicio en cuestión. Ellos se sitúan en la capa más superficial del mismo. Ej: Dropbox.
- Diferencias: en **IaaS** es el desarrollador de las aplicaciones el que debe ocuparse de todo, en **SaaS**, los usuarios ni siquiera tienen acceso al software. Por su parte en **PaaS** pueden gestionar la plataforma, pero no el servidor.

## **7.Diferentes versiones disponibles de Apache Tomcat y sus diferentes versiones de las especificaciones**

### **Tomcat 3.x (distribución inicial)**

- Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
- Recarga de servlets
- Funciones básicas HTTP
- Versión sin soporte

### **Tomcat 4.x**

- Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
- Contenedor de servlets rediseñado como Catalina.
- Motor JSP rediseñado con Jasper.
- Conector Coyote
- Java Management Extensions (JMX), JSP Y administración basada en Struts.
- Versión sin soporte

### **Tomcat 5.x**

- Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- Recolección de basura reducida.
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas.
- Análisis rápido JSP
- Versión sin soporte

### **Tomcat 6.x**

- Implementado de Servlet 2.5 y JSP 2.1

## Tarea 2

- Soporte para *Unified Expression Language 2.1*
- Diseñado para funcionar en Java SE 5.0 y posteriores.
- Soporte para Comet a través de la interfaz CometProcessor.
- Versión sin soporte

### **Tomcat 7.x**

- Implementado de Servlet 3.0 JSP 2.2 y EL 2.2
- Mejoras para detectar y prevenir "fugas de memoria" en las aplicaciones web.
- Limpieza interna de código.
- Soporte para la inclusión de contenidos externos directamente en una aplicación web.

### **Tomcat 8.x**

- Primera versión de Apache Tomcat en soportar las especificaciones de Servlet 3.1, JSP 2.3 y EL 3.0.1.
- Versión sin soporte

### **Tomcat 8.5.x**

- Añade soporte para HTTP/2, OpenSSL para JSSE, TLS virtual hosting y JASPIC 1.1. Creado desde Tomcat 9, siguiendo los retrasos de Java EE 8.

### **Tomcat 9.x**

- Primera versión de Apache Tomcat en soportar las especificaciones de Servlet 4.0.2

## **8.URI, URN Y URL: definición, sintaxis (estructura) y diferentes ejemplos.**

### **URI**

Uniform Resource Identifier. Es una cadena que indica un recurso en Internet. Puede llevar datos adicionales que indiquen cosas como de qué forma acceder al recurso.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

### **URN**

Uniform Resource Name. Indica el nombre de un recurso que debe ser único en todo Internet, además de independiente de su localización.

URN:NID:NSS

urn:isbn:9780007525546

```
<div xmlns:c="http://java.sun.com/jsp/jstl/core" xmlns:field="urn:jsptagdir:/WEB-INF/tags/form/fields" xmlns:jsp="http://java.sun.com/JSP/Page" xmlns:page="urn:jsptagdir:/WEB-INF/tags/form" version="2.0"><code>
</code></div>
```

### **URL**

Uniform Resource Locator. Una URI que indica un recurso en Internet para poder localizarlo. Nos provee de lo necesario para saber dónde está el recurso y de qué forma podemos obtenerlo, indicando el protocolo a usar.

## Tarea 2

<https://obelearningservices.com/cursos/introduccion-programacion/>

2 <https://obelearningservices.com/cursos/json>

3 <ftp://ftp.rediris.es/>

4 <telnet://telnet.wmflabs.org>

## **9.Escalabilidad horizontal: Ventajas y inconvenientes**

### **Ventajas**

- El crecimiento es prácticamente infinito, podríamos agregar cuantos servidores sean necesarios.
- Es posible combinarse con el escalamiento vertical.
- Soporta la alta disponibilidad.
- Si un nodo falla, los demás sigue trabajando.
- Soporta el balanceo de cargas.

### **Inconvenientes**

- Requiere de mucho mantenimiento.
- Es difícil de configurar.
- Requiere de grandes cambios en las aplicaciones (si no fueron diseñadas para trabajar en cluster).
- Requiere de una infraestructura más grande.

## **10.Escalabilidad vertical: Ventajas y inconvenientes.**

### **Ventajas**

- No implica un gran problema para las aplicaciones, pues todo el cambio es sobre el hardware.
- Es mucho más fácil de implementar que el escalamiento horizontal.
- Puede ser una solución rápida y económica (compara con modificar el software).

### **Inconvenientes**

- El crecimiento está limitado por el hardware.
- Una falla en el servidor implica que la aplicación se detenga.
- No soporta la Alta disponibilidad.
- Hacer un upgrade del hardware al máximo pues llegar a ser muy caro, ya que las partes más nuevas suelen ser caras con respecto al rendimiento de un modelo anterior.

## Bibliografía

- <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>
- <https://news.netcraft.com/archives/category/web-server-survey/>
- [https://es.wikipedia.org/wiki/Internet\\_Information\\_Services](https://es.wikipedia.org/wiki/Internet_Information_Services)
- [https://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](https://es.wikipedia.org/wiki/Servidor_HTTP_Apache)
- <https://axarnet.es/blog/saas-paas-iaas>
- [https://es.wikipedia.org/wiki/Tomcat#Historial\\_de\\_versiones](https://es.wikipedia.org/wiki/Tomcat#Historial_de_versiones)
- <https://obelearningservices.com/url-uri-y-urn-explicacion-y-diferencias/>
- <https://www.oscarblancarteblog.com/2017/03/07/escalabilidad-horizontal-y-vertical/>