

TP1 PROG AVANZADA 2025

Mateo Del Valle

Preguntas

Historia de la WWW

1. ¿Cómo surgió el sistema de documentos distribuidos WWW?

El sistema World Wide Web surgió en el CERN como un proyecto para compartir documentos entre investigadores, usando hipertexto y redes distribuidas. Se implementó sobre protocolos de Internet, con un lenguaje de marcado (HTML) y un protocolo de transferencia (HTTP)

2. ¿En qué año se creó la WWW? ¿Quién la creó? ¿Cuál fue el propósito inicial?

La WWW fue creada en **1989** por **Tim Berners-Lee** en el CERN. El propósito inicial fue conectar y compartir información científica de manera sencilla y universal, independiente de la plataforma

3. ¿Sobre qué red funcionaba? ¿Cuáles fueron los componentes iniciales?

Funcionaba sobre la red **Internet (antes ARPANET)**. Los componentes iniciales fueron:

- Protocolo HTTP.
 - Lenguaje HTML.
 - Primer servidor (*httpd*).
 - Primer navegador (WorldWideWeb, luego Nexus)
-

WWW en la actualidad

4. Etapas de evolución y cronología de tecnologías.

- **Web 1.0 (1990–2000):** documentos estáticos en HTML.
- **Web 2.0 (2000–2010):** contenido dinámico, redes sociales, scripting.
- **Web 3.0 (2010–actualidad):** semántica, datos estructurados, IA.

Cronología: HTML (1990), JavaScript (1995), XML y CSS2 (1998), AJAX (2005), HTML5 (2010), WebAssembly y Web3 (2020).

5. ¿Qué es un servidor WWW?

Es un software que atiende solicitudes de clientes y entrega páginas o recursos vía HTTP/HTTPS (ej. Apache, Nginx). Se ejecuta sobre un sistema operativo y utiliza TCP/IP para la comunicación

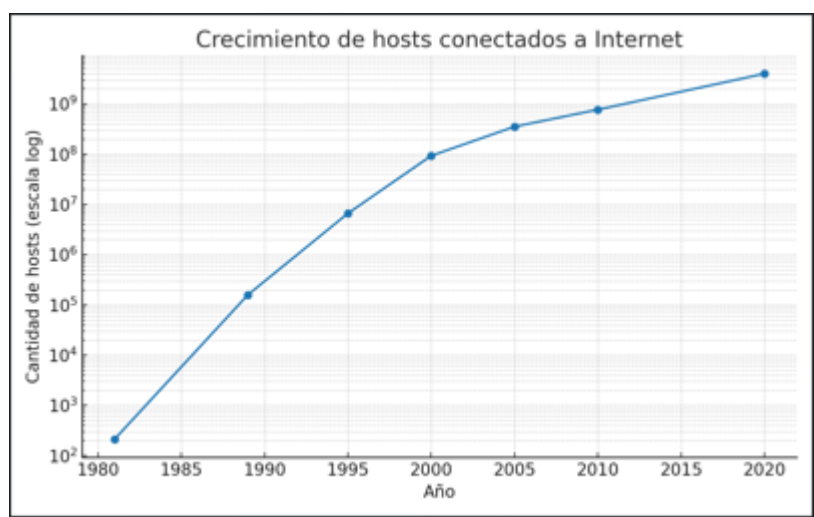
6. ¿Qué es un cliente web?

Un cliente web es el programa (navegador) que solicita recursos al servidor, los recibe e interpreta. Envía peticiones HTTP y renderiza el contenido (HTML, CSS, JS). Ejemplos: Chrome, Firefox

7. ¿Cómo está compuesto el sistema WWW actual?

- **Clientes:** navegadores, apps móviles.
- **Servidores:** web, bases de datos.
- **Protocolos:** HTTP/HTTPS, DNS, TCP/IP.
- **Lenguajes:** HTML5, CSS, JS, PHP, Python.
- **Servicios distribuidos:** APIs, servicios web.

8. Diagrama de crecimiento de hosts.



9. Diagrama conceptual de la composición de la Web.



10. Arquitectura del sistema distribuido de documentos.

Modelo cliente/servidor distribuido:

- Cliente: navegador.
- Servidor: entrega recursos.
- Protocolo: HTTP/TCP.
- Documentos: identificados por URL, escritos en HTML.
- Hipervínculos: interconexión de recursos en distintos servidores.

Tecnologías, Servidores y Clientes

11. ¿Cómo funciona el sistema WWW?

El WWW funciona bajo una arquitectura distribuida cliente/servidor:

1. El cliente (navegador) envía una solicitud HTTP a un servidor.
2. El servidor procesa la solicitud, busca el recurso y responde con un documento (ej. HTML).
3. El cliente interpreta y muestra el recurso.

12. ¿Qué es HTML? ¿De dónde proviene?

HTML (*HyperText Markup Language*) es un lenguaje de marcado que estructura documentos hipertexto. Fue creado por Tim Berners-Lee en 1991 en el CERN como parte de la primera implementación de la web

13. ¿Qué es una página web estática, dinámica y activa?

- **Estática:** documentos HTML fijos, su contenido no cambia (ej. sitios informativos).
 - **Dinámica:** generada en tiempo real por el servidor (ej. PHP, JSP).
 - **Activa:** incluye interacción en el cliente con scripts como JavaScript (ej. validación en formularios).
-

14. ¿Qué es una URL? ¿Qué función cumple en la Web? ¿Qué es un servidor de archivos basado en URL?

- Una **URL** (Uniform Resource Locator) es un identificador único que señala dónde está un recurso y cómo accederlo.
 - Su función es direccionar documentos en la web.
 - Un servidor de archivos basado en URL responde a estas direcciones entregando recursos (ej. servidor HTTP).
-

15. ¿Dónde y cómo se ejecuta el lenguaje HTML?

HTML no se "ejecuta", sino que se **interpreta** por el navegador en el lado del cliente. El servidor solo lo entrega al cliente. El navegador analiza el documento y lo presenta gráficamente al usuario

16. ¿Qué es un lenguaje script en la tecnología WWW?

Un script es un lenguaje interpretado que se incrusta en las páginas web para dar dinamismo. Puede ejecutarse en el **cliente** (JavaScript) o en el **servidor** (PHP, ASP). Permite validaciones, interactividad, acceso a bases de datos

17. ¿Qué es un plugin? ¿Qué es una cookie?

- **Plugin:** extensión que añade funcionalidades extra al navegador (ej. Flash, PDF Reader).
 - **Cookie:** archivo de texto que almacena datos de sesión del usuario en el cliente, permitiendo recordar preferencias o estados de login.
-

18. ¿Qué es JavaScript? ¿Dónde y quién lo ejecuta? ¿Cómo nació?

JavaScript es un lenguaje de scripting que se ejecuta en el navegador del cliente. Nació en 1995 por Netscape (Brendan Eich) como complemento a HTML y CSS para generar interactividad. Hoy es estándar de facto en la web

Tecnologías, Servidores y Clientes

19. ¿Qué es un Applet de Java? ¿Cómo puede un browser ejecutar código Java?

Un **Applet de Java** es un pequeño programa escrito en Java que se ejecuta dentro de un navegador web, originalmente mediante una *Java Virtual Machine (JVM)* embebida. El browser necesita un plugin o máquina virtual instalada localmente para poder ejecutar el código Java descargado. Actualmente está en desuso por razones de seguridad y soporte

20. ¿Cómo funciona la tecnología Flash de Macromedia?

Flash fue una tecnología multimedia que permitía reproducir animaciones e interactividad en la web a través de un plugin en el navegador. Utilizaba un lenguaje propio (ActionScript) para controlar objetos. Permitía contenido interactivo antes de la masificación de HTML5 y JavaScript, pero fue abandonado por problemas de seguridad y falta de soporte

21. ¿Para qué se utiliza la tecnología CGI? ¿Qué lenguajes se pueden usar en scripts CGI?

CGI (*Common Gateway Interface*) es una tecnología que conecta el servidor web con programas externos que generan contenido dinámico. Permite que al recibir una petición, el servidor ejecute un script y devuelva resultados al cliente. Los scripts CGI se pueden programar en **Perl, Python, C, PHP, Shell script**, entre otros

22. ¿Qué es un "form" HTML? ¿Para qué se utiliza?

Un formulario HTML es un conjunto de controles de entrada (textos, botones, menús) que permiten al usuario enviar datos al servidor. Se utilizan para registro de usuarios, búsquedas, encuestas, compras en línea, etc. Los datos se envían mediante métodos HTTP (GET o POST) ([W3C, sección *HTML Forms*]).

23. ¿Qué son y cómo funcionan las tecnologías ASP, PHP y JSP?

- **ASP (Active Server Pages):** tecnología de Microsoft para generar páginas dinámicas desde el servidor usando VBScript o C#.
- **PHP:** lenguaje de scripting del lado del servidor, ampliamente usado en sitios dinámicos.
- **JSP (Java Server Pages):** extensión de Java que permite incrustar código Java en páginas web, ejecutado en un servidor con soporte para Servlets.

Las tres permiten generar HTML dinámico en tiempo real, integrándose con bases de datos y lógica de negocio

24. ¿Qué es DHTML? ¿Y XHTML?

- **DHTML (Dynamic HTML):** combinación de HTML, CSS y JavaScript para crear páginas interactivas sin necesidad de recargar.
 - **XHTML (eXtensible HTML):** reformulación de HTML como aplicación XML, más estricta en su sintaxis para asegurar compatibilidad e interoperabilidad.
-

25. ¿Qué es XML? Explique detalladamente.

XML (*eXtensible Markup Language*) es un lenguaje de marcado diseñado para describir y transportar datos de forma estructurada y legible tanto por humanos como por máquinas. No define cómo mostrar la información, sino cómo organizarla mediante etiquetas personalizadas. Se utiliza en servicios web, configuración, intercambio de datos entre sistemas

26. ¿Qué son los Servicios Web? Dé un ejemplo de aplicación.

Un **Servicio Web** es una aplicación accesible a través de protocolos estándar de Internet (HTTP, XML, SOAP, REST) que permite la comunicación entre diferentes sistemas. Ejemplo: un servicio de clima que devuelve datos en XML/JSON al consultar con una URL, usado en aplicaciones móviles o páginas web

URL's, Dominios y Direcciones

27. ¿Qué es una URL? ¿Qué función cumple en la Web? ¿Cuál es la estructura de una URL?

Una **URL (Uniform Resource Locator)** es la dirección única que identifica un recurso en la Web y especifica cómo acceder a él.

- **Función:** permite localizar y recuperar documentos y recursos a través de la red.
- **Estructura básica:**

```
protocolo://host[:puerto]/ruta/recurso?parámetros#fragmento
```

Ejemplo: <https://www.ejemplo.com:80/docs/index.html?id=10#seccion2>

28. ¿Qué es un servidor de archivos basado en URL?

Es un servidor que responde a solicitudes de clientes entregando archivos identificados mediante URLs. El protocolo HTTP es el más común, aunque también existen servidores FTP o de streaming que trabajan con localizadores uniformes

29. ¿Cuál es el formato de una URL en IPv6?

En IPv6, la dirección se encierra entre corchetes `[]` dentro de la URL.

Ejemplo:

```
http://[2001:db8::1]/index.html
```

Esto asegura que los dos puntos del formato IPv6 no se confundan con el separador de puerto

30. ¿Qué es un Dominio www? ¿Qué relación tiene con una URL?

Un **dominio** es el nombre jerárquico que identifica un recurso dentro del sistema DNS (ej. www.google.com). En una URL, el dominio forma parte del **host** y traduce la dirección numérica IP a un nombre fácil de recordar

31. ¿Cuál es la función del DNS en la WEB?

El **Domain Name System (DNS)** traduce los nombres de dominio (ej. www.ejemplo.com) en direcciones IP que identifican servidores en la red. Sin DNS, los usuarios tendrían que recordar direcciones IP numéricas en lugar de nombres simbólicos

Protocolos utilizados en la WWW

32. ¿Cuál es la relación entre HTML y HTTP?

- **HTML** define la estructura del contenido (hipertexto).
- **HTTP** es el protocolo que transporta ese contenido entre cliente y servidor.

La relación es directa: HTML viaja dentro de los mensajes HTTP para ser interpretado por el navegador

33. ¿Cuáles son las versiones de HTTP y cuáles sus diferencias?

- **HTTP/0.9 (1991)**: solo permitía transferir texto sin cabeceras.
- **HTTP/1.0 (1996)**: soporte para cabeceras, tipos MIME, códigos de estado.
- **HTTP/1.1 (1997)**: conexiones persistentes, chunked transfer, más métodos.
- **HTTP/2 (2015)**: binario, multiplexación, compresión de cabeceras.
- **HTTP/3 (2020)**: basado en QUIC (UDP), más rápido y seguro.

34. ¿Qué es MIME?

MIME (Multipurpose Internet Mail Extensions) es un estándar para indicar el tipo de contenido en un mensaje. En HTTP, se utiliza en la cabecera `Content-Type` para especificar si un recurso es `text/html`, `image/png`, `application/json`, etc.

35. Explique cómo es una solicitud HTTP.

Una solicitud HTTP está compuesta por:

1. **Línea de petición**: método (ej. GET, POST), URL, versión.

```
GET /index.html HTTP/1.1
```

2. **Cabeceras (headers)**: metadatos (ej. Host, User-Agent, Accept).
3. **Cuerpo (body)**: datos opcionales (ej. formularios enviados con POST).

36. ¿Cuáles son los métodos de HTTP/1? Enumere y describa.

- **GET**: solicita un recurso.
- **POST**: envía datos al servidor.
- **HEAD**: igual que GET pero solo cabeceras.

- **PUT:** sube un recurso al servidor.
 - **DELETE:** elimina un recurso.
 - **OPTIONS:** consulta operaciones permitidas sobre un recurso.
 - **TRACE:** devuelve la petición recibida (debug).
-