

# Arquitectura Web

Curso 2020/21

Javier Albert Seguí

# Indice

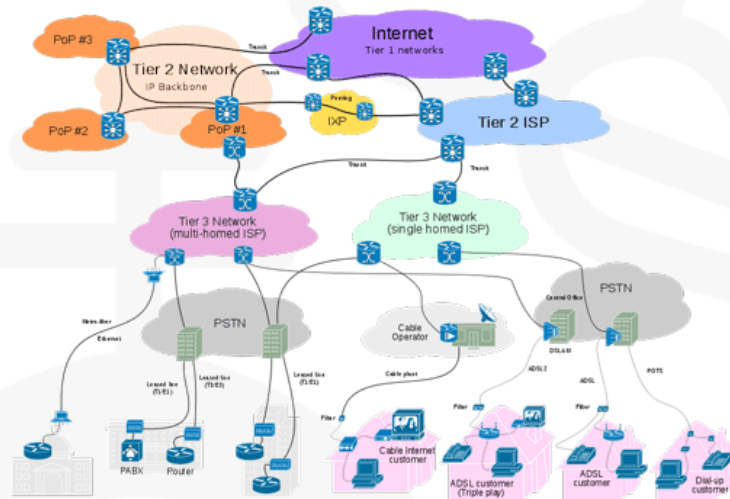
- Evolución y funcionamiento de Internet
- Servicios y protocolos de Internet
- Protocolo HTTP
- Desarrollo de aplicaciones Web
- Arquitectura Web

# Evolución de Internet

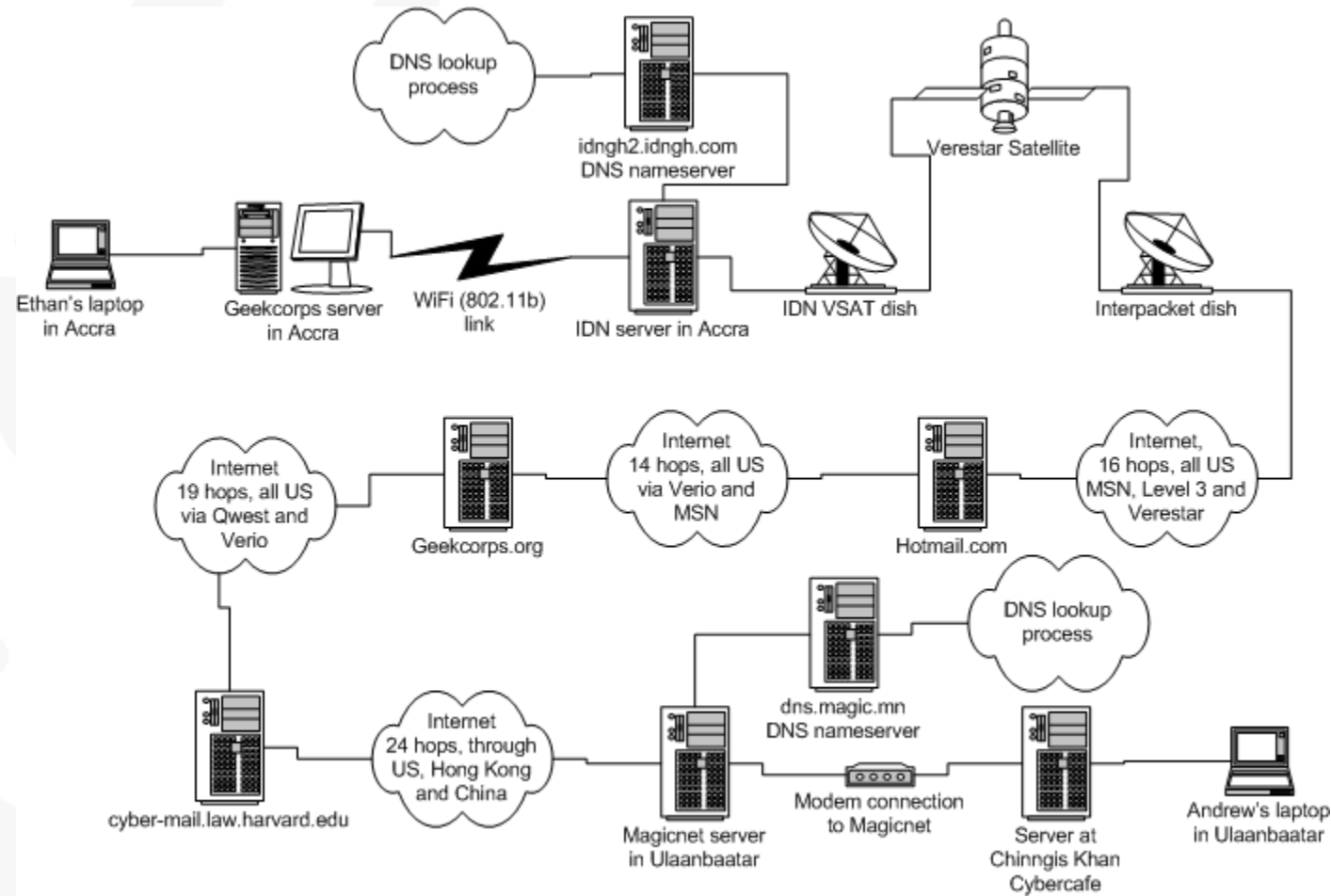
- Internet no es mas que una red de ordenadores de tamaño mundial, que usa como protocolo de comunicaciones **TCP/IP**
- **Internet** dio sus primeros pasos en Estados Unidos a finales de los años 60. Como evolución de la red del DoD para comunicación en caso de guerra, **ARPAnet**.

# Como funciona Internet

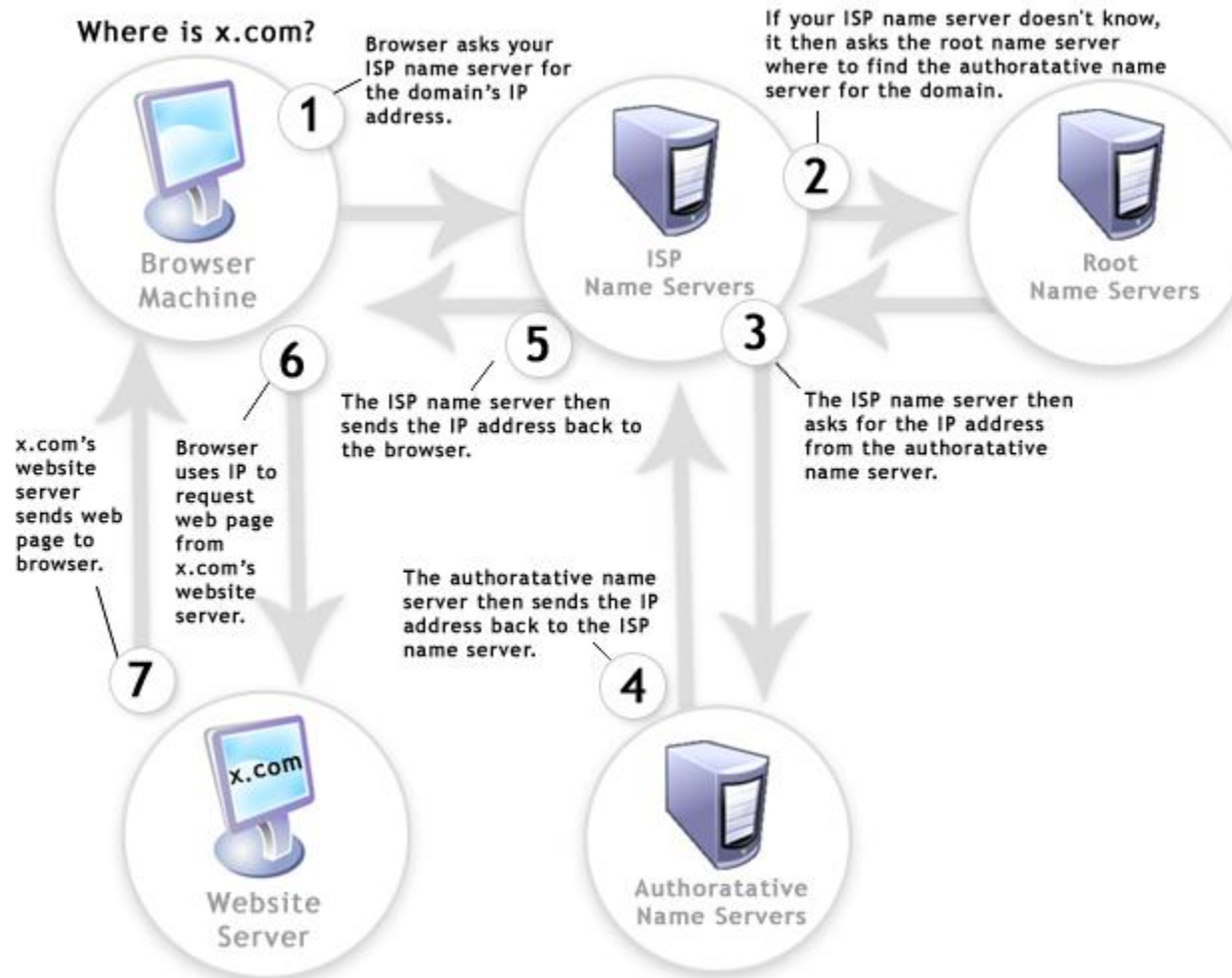
- Internet responde a una arquitectura **Cliente/Servidor**
- En el momento que utilizamos alguno de los servicios que Internet nos ofrece, se pone en marcha un extenso entramado de maquinas y aplicaciones que hacen posible ese funcionamiento.



# Como funciona Internet



# Como funciona Internet



# Protocolos de Internet

- Los Servicios a los que podemos acceder en internet son muy diversos.
- Definiremos **Servicio** como un conjunto de programas y utilidades que nos permiten realizar una determinada tarea.
- Servicios:
  - FTP ( File Transfer Protocol)
  - DNS
  - Gopher
  - News
  - Correo electrónico
  - Mensajería instantánea
  - Redes P2P
  - **Web**

# Protocolos de Internet

- El proyecto **World Wide Web** nació en respuesta a la necesidad que la comunidad científica internacional tenía de nuevos sistemas de distribución de la información.
- Este fue uno de los objetivos que se planteó **Tim Berners-Lee** (ingeniero británico) cuando en 1989 presentó a sus superiores del **CERN** la propuesta original para el proyecto World Wide Web. (<http://www.w3c.org>)
- Desde el punto de vista del usuario, la Web consiste en un enorme conjunto, a nivel mundial, de documentos llamados páginas. Cada página puede contener vínculos (enlaces) con otras páginas relacionadas en cualquier lugar del mundo. Podemos afirmar que es un gran almacén o repositorio de información.



# Protocolos de Internet

- Cada servidor Web tiene un demonio (proceso) que escucha en el puerto TCP80, esperando conexiones entrantes. Tras establecerse una conexión, el cliente envía una solicitud, el servidor genera una respuesta y finalmente se libera la conexión.
- El servicio Web responde a un modelo **Cliente/Servidor**, donde los clientes demandan “hipertextos” a los servidores. La web es un sistema hipermedia distribuido de acceso interactivo para los usuarios.
  - Hipertexto = Documentos con “punteros” a otros documentos.
  - Hipermedia = Hipertexto + información multimedia (imágenes, sonidos, vídeos,...)

# El “lenguaje” de la Web

- El protocolo de Comunicaciones es HTTP – HyperText Transfer Protocol
- El lenguaje para poder crear documentos válidos es HTML – HyperText Markup Language
- El esquema de direcciones que permiten localizar Recursos en Internet es URL – Universal Resource Locator
- Las “herramientas” que interpretan el lenguaje HTML y lo presentan a los usuarios son los Navegadores

# Protocolo HTTP

- Es la base sobre la que se sustenta la Web
- Es un protocolo del nivel de aplicación para sistemas de información multimedia distribuidos.
- Es un protocolo no orientado a estado, que permite el intercambio de información.
- Al ser un protocolo sin estado, se hace necesario que al crear aplicaciones en entornos web se necesite mantener el estado de las mismas a través de Sesiones.
- Basado en las RFC1945/2616/2774/7540.....

# Protocolo HTTP

- Tiene las siguientes propiedades:
  - Se asienta sobre el paradigma **solicitud/respuesta**
  - La comunicación utiliza el protocolo **TCP/IP**
  - El puerto por defecto es es 80, pero puede utilizarse cualquier otro.
  - Los mensajes/transacciones son en texto plano lo que lo hace legible y fácil de procesar y depurar.
  - Está abierto a nuevos tipos de datos, utiliza tipos MIME (Multipart Internet Mail Extension) para determinar el tipo de datos que transporta.
  - Tiene un esquema de direccionamiento comprensible:  
**servicio://host:puerto/recurso**

# Protocolo HTTP

## ○ Cliente

- Get / HTTP/1.1
- Host: [www.example.com](http://www.example.com)
- User-Agent: Lynx/2.2 libwww/2.14

## ○ Servidor

- HTTP/1.1 200 OK
- Date: Mon, 23 May 2005 22:38:34 GMT
- Content-Type: text/html; charset=UTF-8
- Content-Length: 155
- Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
- Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
- ETag: "3f80f-1b6-3e1cb03b"
- Accept-Ranges: bytes
- Connection: close

# Protocolo HTTP

## ○ Comandos HTTP :

- GET Petición de recurso.
- POST Petición de recurso pasando parámetros.
- HEAD Petición de datos sobre recurso.
- PUT Creación o envío de recurso.
- DELETE Eliminación de recurso.
- TRACE Devuelve al origen la petición tal como se ha recibido en el receptor, para depurar errores.
- OPTIONS Sirve para comprobar las capacidades del servidor.
- CONNECT Reservado para uso en servidores intermedios capaces de funcionar como túneles.
- PATCH: Aplica modificaciones parciales a un recurso ya cargado.

# Protocolo HTTP

## ○ Respuestas:

- Informational 1XX – Petición recibida, continúa proceso.
- Successful 2XX – Petición procesada correctamente
- Redirection 3XX – Petición redirigida
- Client Error 4XX – No se puede procesar la petición porque es incorrecta
- Server Error 5XX – El servidor ha fallado en el proceso de la petición.

# Protocolo HTTP

## ○Cabecera:

- Son metadatos que se envían en las transacciones de petición o respuesta para proporcionar información esencial sobre el mensaje en curso.
- Cada cabecera es especificada por un nombre de cabecera seguido por dos puntos, un espacio en blanco y el valor de dicha cabecera seguida por un retorno de carro.
  - Host: `www.prueba.com`
  - Accept: `www/source`
  - Accept: `text/html`
  - Accept: `image/gif`
  - User-Agent: `Lynx/2.2 libwww/2.14`



# Desarrollo de aplicaciones Web

- Con la introducción de Internet y de la Web en concreto, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi cualquier sitio.
- Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizarse en la Web.

# Desarrollo de aplicaciones Web

## ○ El principio....

- CGI (Common Gateway Interface) es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME.
- Las aplicaciones CGI fueron una de las primeras prácticas de crear contenido dinámico para las páginas web.
- En una aplicación CGI, el servidor web pasa las solicitudes del cliente a un programa externo. Este programa puede estar escrito en cualquier lenguaje que soporte el servidor, aunque por razones de portabilidad se suelen usar lenguajes de script. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional.

# Desarrollo de aplicaciones Web

## ○Actualidad:

- Microsoft ( .NET + ASP + RAZOR + BLAZOR.....)
- Java ( Servlets, JSP.....)
- Otros ( PHP, NodeJS.... )
- Cada vez hay mas frameworks que ayudan a los programadores a un Desarrollo mas rápido de aplicaciones para la web ( JSF, Struts, Spring, Symfony....)

# Arquitectura Web

- Suele presentar un esquema de 3 niveles:
  - Nivel 1: Capa de presentación.
  - Nivel 2: Capa de lógica de negocio.
  - Nivel 3: Capa de acceso a datos.