Arquitectura Web

Desarrollo de aplicaciones web

Agenda

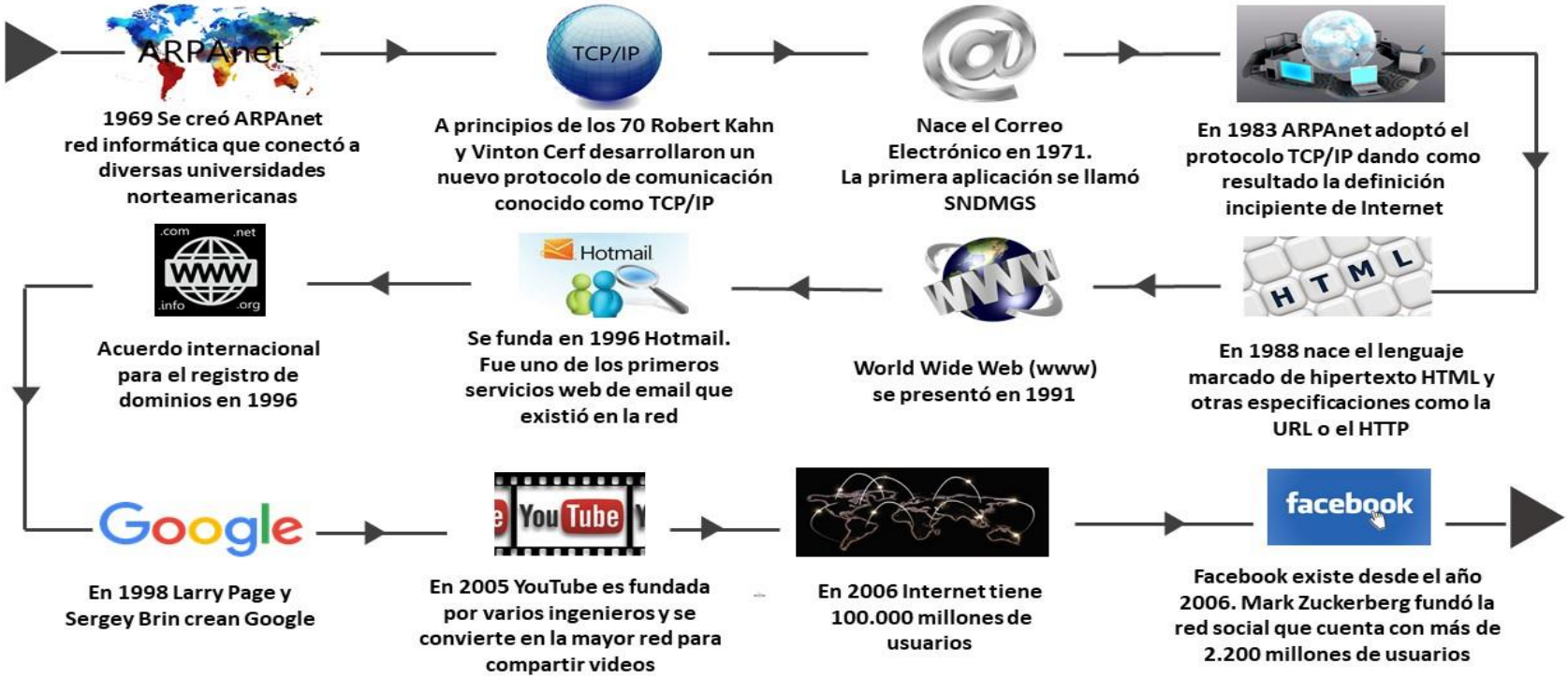
- Arquitectura Web en el Cliente-Servidor.
- Generalidades del Protocolo HTTP.
  - HTTP Headers.
  - HTTP Request.
  - Métodos HTTP.
- HTTP Response Status.
- Controles de acceso en HTTP (Cors).

Objetivos:

- Conocer arquitecturas usadas en el desarrollo de aplicaciones web
- Aplicar protocolo http en el desarrollo de aplicaciones web

Historia de Internet

HISTORIA DE INTERNET: GRANDES HITOS



LA WEB

- La Web se puede considerar como una plataforma o “sistema operativo” en el cual los recursos están distribuidos en la Red y están siendo extendidos en todo momento con posibilidades ilimitadas.

Funcionamiento de un Servidor Web

- Un servidor web es un software que forma parte del servidor y tiene como misión principal devolver información (páginas) cuando recibe peticiones por parte de los usuarios.

Componentes semánticos de la Web

- URI: Uniform Resource Identifier.
  - Identifica los recursos web para su acceso y manipulación.
- HTML: HyperText Markup Language.
  - Lenguaje de marcas.
  - Provee una representación estándar de los documentos hipertexto en formato ASCII.
  - Permite formatear texto, integrar imágenes, referenciar otros documentos, etc.
- HTTP: Hypertext Transfer Protocol.
  - Protocolo que permite a los componentes web (cliente, servidores, etc) comunicarse de una forma estándar y bien definida.
  - Define el formato y el significado de los mensajes intercambiados entre componentes web.

Arquitectura de aplicaciones web

- Las aplicaciones web utilizan lo que se conoce como clientes livianos (light clients) los cuales no ejecutan demasiadas labores de procesamiento para la ejecución de la aplicación misma.
- Desde el punto de vista de la arquitectura se distinguen dos lados:
  - El cliente, donde se encuentra el usuario final utilizando la aplicación por medio de un navegador (como Internet Explorer o Mozilla Firefox). A través de este cliente web, el usuario interactúa con la aplicación localizada al otro lado, en
  - El servidor, que es donde residen realmente los datos, reglas y lógica de la aplicación.

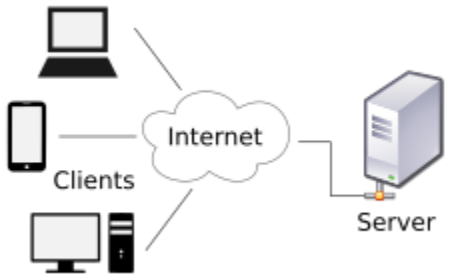
ARQUITECTURA DE TRES NIVELES

- En la arquitectura en tres niveles existe un nivel intermedio. Esto significa que la arquitectura generalmente está compartida por:
  - Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación.
  - El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
  - El servidor de datos, que proporciona al servidor de aplicaciones los datos que éste le solicitó.

ARQUITECTURA DE TRES NIVELES

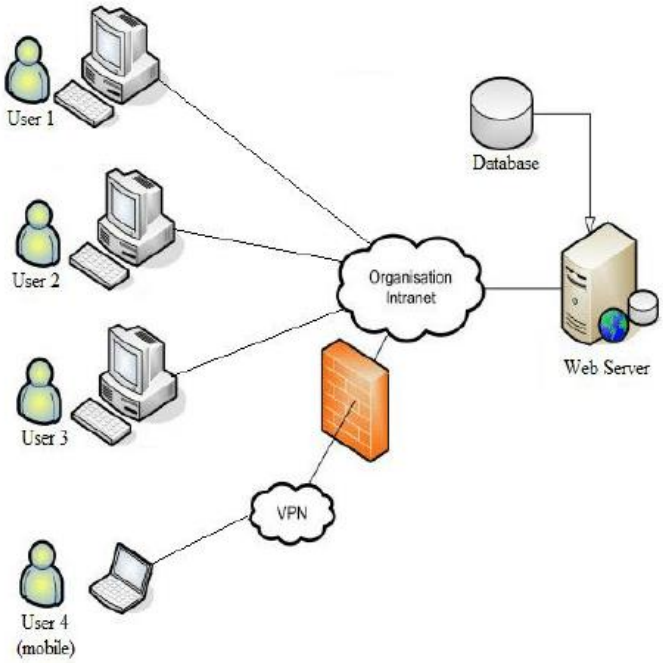
Figura 1.1. *Arquitectura de 3 niveles*

Arquitectura Web en el Cliente-Servidor



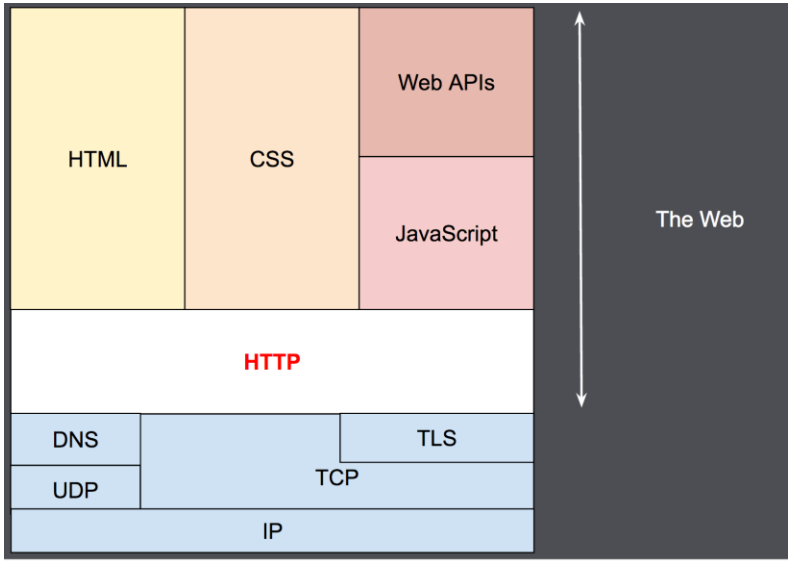
¿Qué es la Arquitectura Cliente-Servidor?

- Un modelo de interacción donde dos tipos de software cooperan para proveer un servicio:
- **Cliente:** Inicia la solicitud y recibe la respuesta.
- **Servidor:** Procesa la solicitud y envía la respuesta.



¿Qué es el Protocolo HTTP?

- Protocolo sin estado, basado en texto, que define cómo los navegadores y servidores web se comunican.
- Características:
  - **Sin estado**: Cada solicitud y respuesta son independientes.
  - **Basado en texto:** Los mensajes se codifican en formato de texto plano.
  - **Versátil**: Se utiliza para diversos servicios web, como la transferencia de páginas web, imágenes y datos.



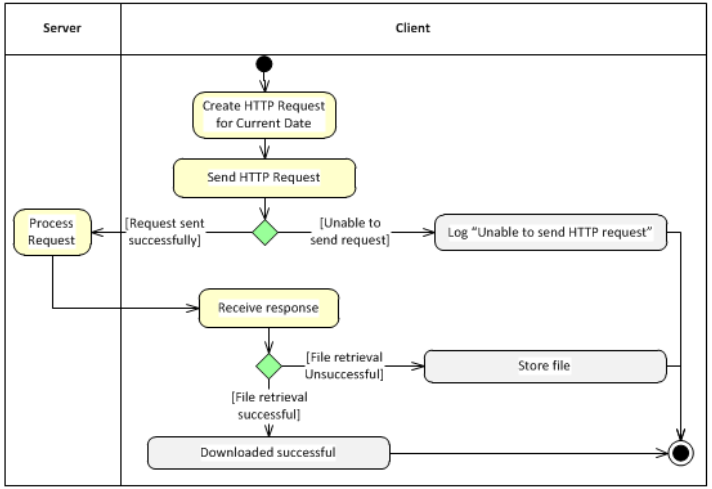
Encabezados HTTP

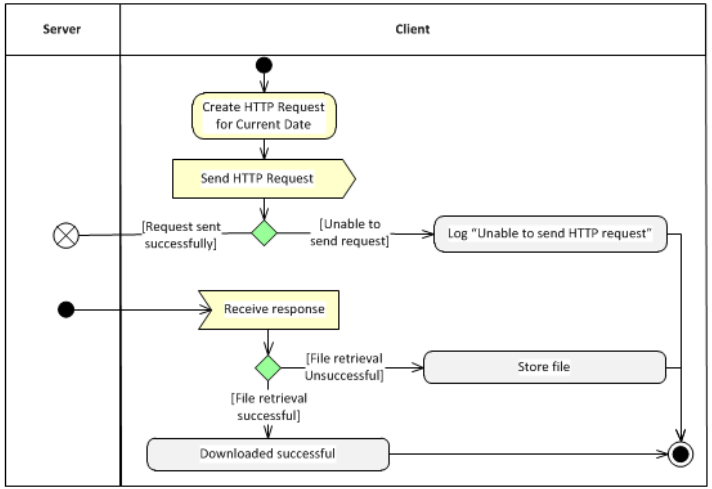
- Información adicional que se incluye en cada solicitud y respuesta HTTP.
- Propósito:
  - **Proporcionar metadatos sobre la solicitud o respuesta.**
  - **Especificar el tipo de contenido, la longitud del mensaje, el idioma y otros detalles.**
- Ejemplos de encabezados comunes:
  - **Host:** Identifica el nombre de dominio del servidor.
  - **Content-Type:** Especifica el tipo de contenido de la respuesta, como HTML, CSS o JSON.
  - **Accept:** Indica los tipos de contenido que el cliente puede aceptar.
  - **Authorization:** Contiene información de autenticación para acceder a recursos protegidos.

The request data	
RequestPacketIndex	index of the packet at all received packet list
Req_HTTPMethod	Requests a web application override the method specified in the request
Req_RequiredURL	The requested URL
Req_HTTPVersion	Used HTTP Version
Req_Host	The domain name of the server
Req_UserAgent	The user agent string of the user agent
Req_Accept	Content-Types that are acceptable for the response
Req_AcceptLanguage	List of acceptable human languages for response
Req_AcceptEncoding	List of acceptable encodings
Req_Cookie	An HTTP cookie previously sent by the server
Req_Connection	Control options for the current connection and list of hop-by-hop request fields
Req_Pragma	Implementation-specific fields that may have various effects anywhere along the request-response chain
Req_CacheControl	Tells all caching mechanisms from server to client whether they may cache this object
The response data	
Response-PacketIndex	index of the packet at all received packet list
Res_StatusCode	CGI header field specifying the status of the HTTP response

Solicitud HTTP

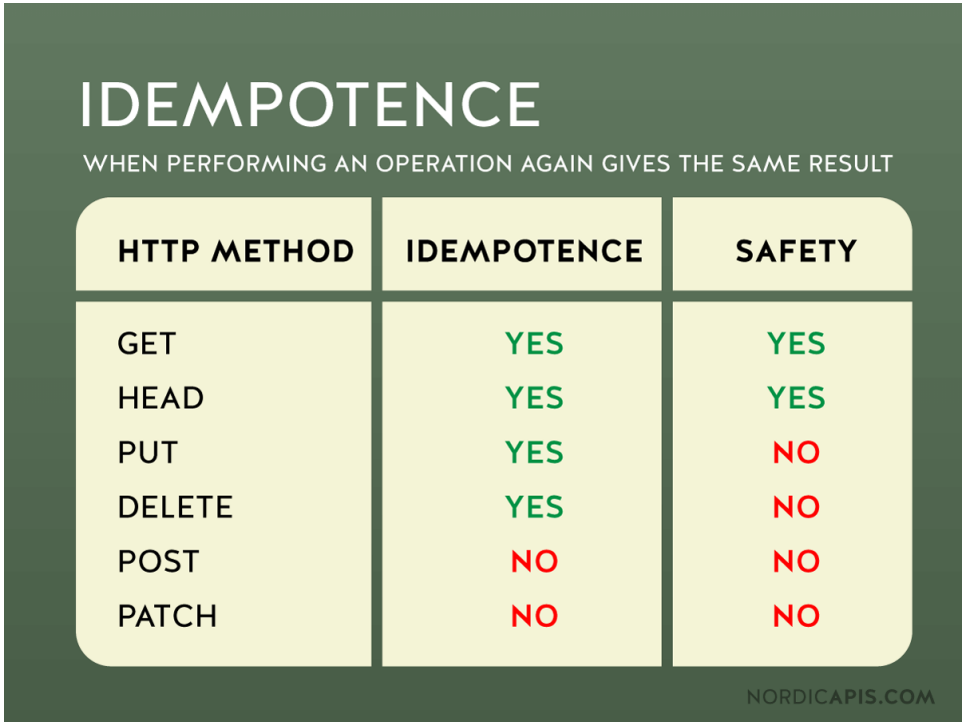
- Mensaje enviado por el cliente al servidor para iniciar una acción.
- Estructura:
  - **Método HTTP:** Indica la acción que se desea realizar, como GET, POST, PUT o DELETE.
  - **URL:** Especifica la ubicación del recurso que se solicita.
  - **Encabezados:** Proporcionan metadatos sobre la solicitud.
  - **Cuerpo:** Opcional, contiene datos adicionales para el servidor.





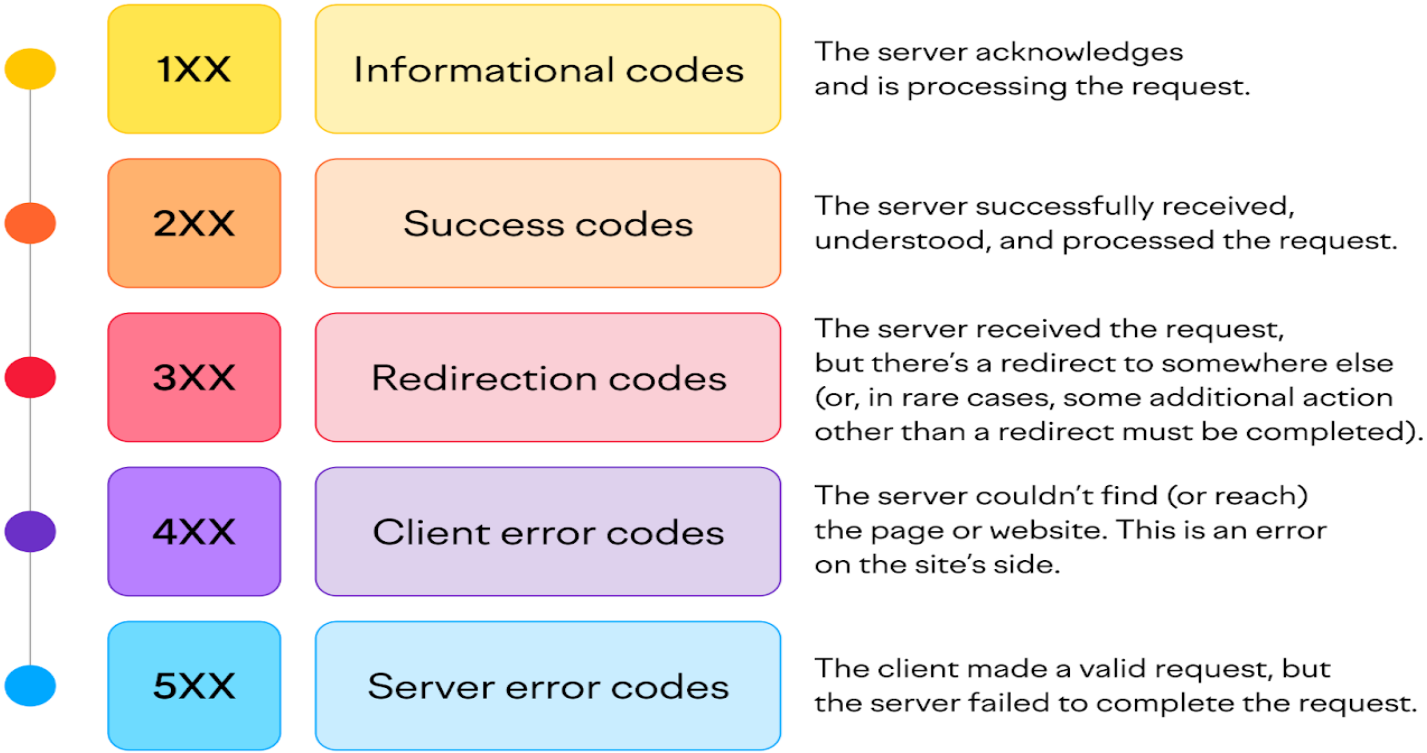
Métodos HTTP

- Verbos que indican la acción que se desea realizar en un recurso.
- Métodos comunes:
  - **GET:** Recupera información de un recurso.
  - **POST:** Envía datos al servidor para crear o actualizar un recurso.
  - **PUT:** Actualiza un recurso existente con nuevos datos.
  - **DELETE:** Elimina un recurso.
  - **OPTIONS:** Obtiene información sobre los métodos HTTP admitidos por un recurso.



Códigos de estado HTTP

Códigos numéricos que el servidor envía al cliente



Bibliografía

- **BÁSICA**:
  - **BB1** Purewal, Semmy. (2014). Learning web app development. (First edition.;). EEUU: O'Reilly Media. ISBN-10:1449370195, ISBN-13: 9781449370190
  - **BB2** Robbins,Jennifer. (2018). Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. (Fifth Edition). EEUU: O'Reilly Media. ISBN-10: 1491960205,ISBN-13:9781491960202
- **COMPLEMENTARIA**:
  - **BC1** Flanagan, David. (2011). JavaScript: The Definitive Guide:Activate Your Web Pages. (Paperback; 2011-05-03). EEUU:O'Reilly Media. ISBN-10: 0596805527, ISBN-13: 9780596805524