



Universidad Politécnica de Madrid

Introducción al desarrollo de aplicaciones web

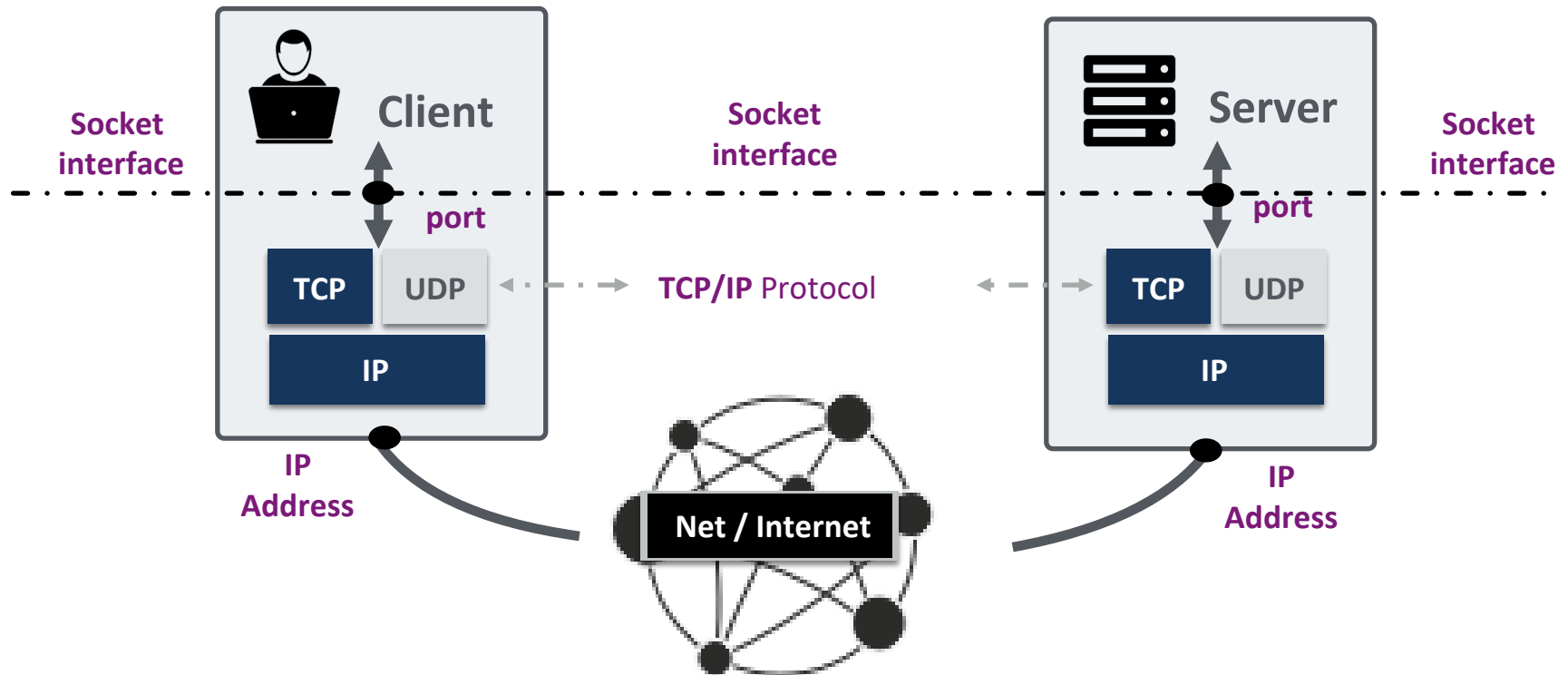
Enrique Barra Arias
Álvaro Alonso González

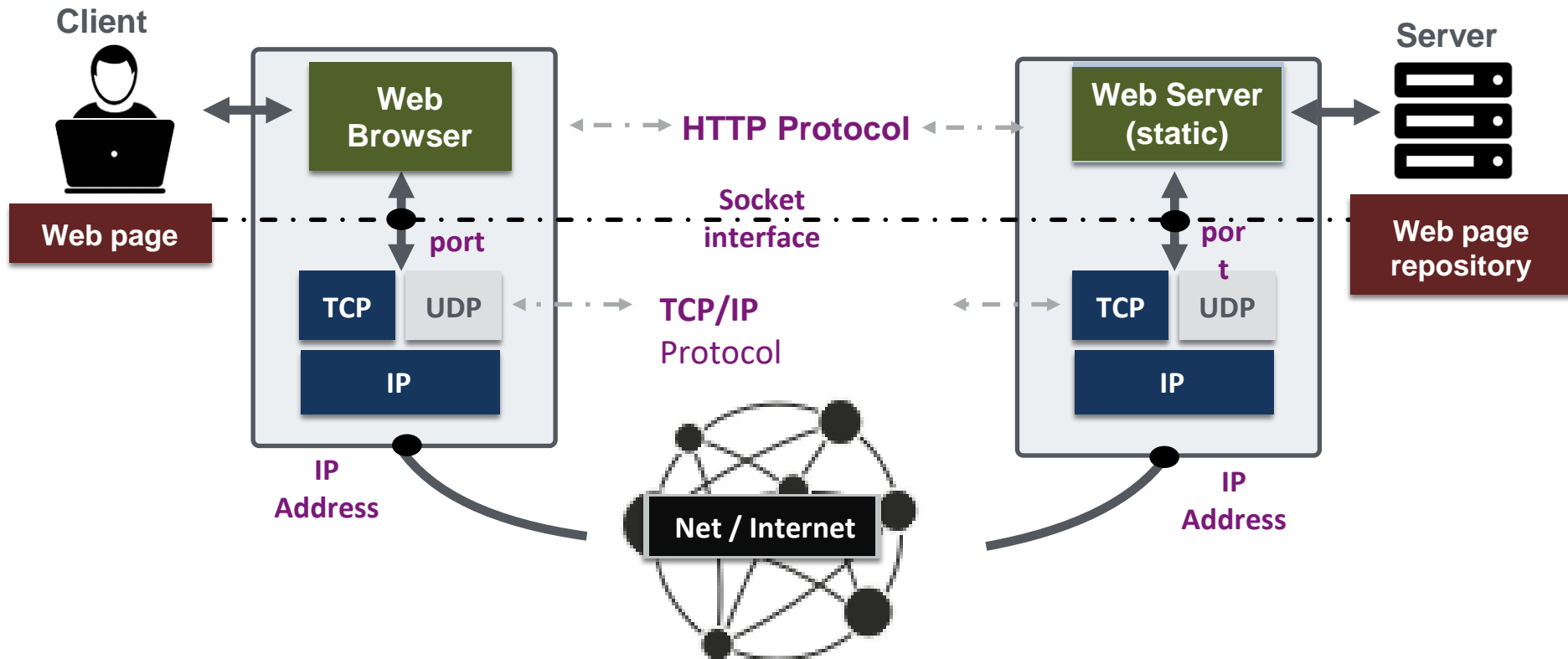
- Internet comenzó con Arpanet en enero de 1983
- Internet conecta ordenadores a la red usando el protocolo TCP/IP
- TCP/IP soporta aplicaciones **cliente-servidor** a través de una interfaz de sockets
- Las primeras aplicaciones sobre Internet fueron:
 - telnet (terminal virtual), ftp (file transfer), email, ...
- El modelo **client-server** es una estructura de aplicación distribuida que divide las tareas a realizar entre proveedores de recursos o servicios, llamados **servidores**, e interesados en estos servicios o productos, llamados **clientes**

- Tim Berners Lee propuso en 1989 una nueva aplicación: la World Wide Web o la Web
- La Web es un espacio de información donde encontramos documentos y otros recursos unívocamente identificados por su Uniform Resource Locator (URL), interconectados entre sí por hipervínculos y accesibles a través de Internet
- Aplicación cliente (navegador) <-> aplicación servidor (servidor web)
- La Web es el depósito de contenido que Internet necesitaba
- Transforma Internet en una red de distribución de contenido

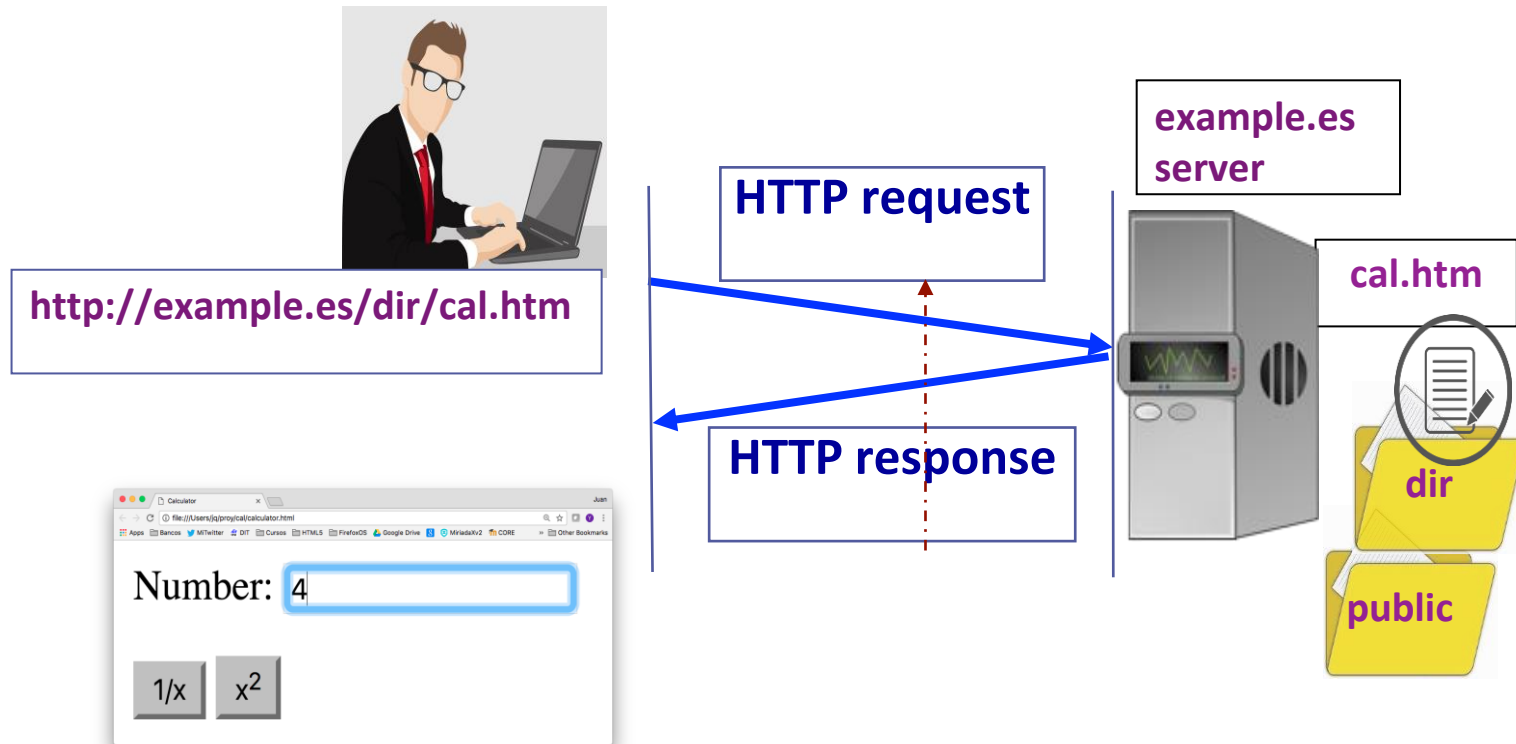
Tim Berners Lee desarrolló/unió 3 tecnologías esenciales:

1. un sistema de identificadores globalmente únicos para los recursos en la Web y en otros lugares, el identificador universal de documentos (UDI), más tarde conocido como localizador uniforme de recursos (URL) e identificador uniforme de recursos (URI)
2. el lenguaje de publicación HyperText Markup Language (HTML)
3. el Protocolo de Transferencia de Hipertexto (HTTP)



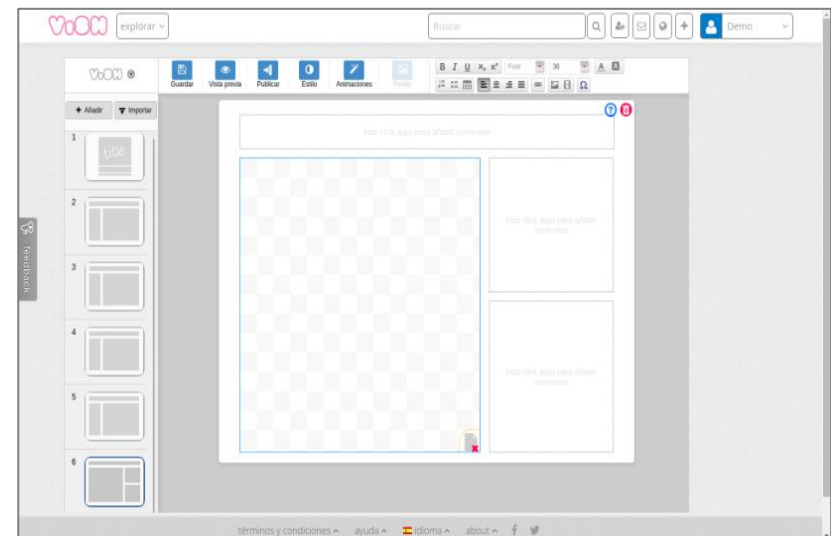
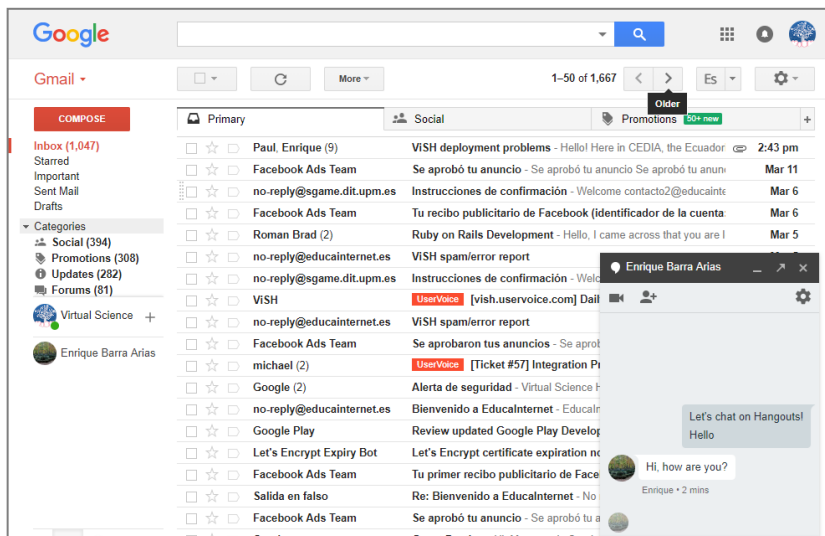


- Servidor web estático
 - Programa que sirve archivos solicitados por los clientes
- Cliente web
 - Programa que solicita las páginas web a un servidor y las presenta al usuario
 - El URL identifica al recurso Web
 - Por ejemplo: <http://example.es/dir/cal.html>
 - **http**: protocolo para acceder al servidor (HTTP GET)
 - **example.es**: dirección del servidor que aloja la página
 - **/dir/cal.html**: ruta del fichero (o pág. web) en el directorio del servidor
- Transacción HTTP vista por el cliente
 - El cliente establece la conexión TCP con el servidor
 - El servidor envía la respuesta HTTP con el recurso a mostrar
 - El cliente recibe a través de la conexión HTTP la respuesta
 - El servidor cierra la conexión TCP



- HTTP -> HTTP1.1, HTTPS (HTTP + SSL), HTTP2, ...
 - Nuevos protocolos para nuevos tipos de apps
- JavaScript -> ES6 y ES7
- HTML/CSS -> HTML5, CSS3
- Cliente web (navegador) -> Evolucionan constantemente incluyendo nuevos estándares, pestañas, extensiones...
- Servidor web -> dinámicos, conectados a BBDD...

- 90s
 - Poco JS. Páginas estáticas.
 - Lógica del lado del servidor sólo
- 2000s
 - AJAX
- 200X
 - SPA



Páginas estáticas



➤ Proceso

- El cliente (navegador) solicita información al servidor, la página web (.html) especificada en la barra de dirección o a través de un link
- El servidor envía esa página
- El cliente solicita el resto de recursos de la página (CSS, JS, imágenes, vídeos, ...) y muestra (dibuja) la página

➤ Características

- La lógica está en el servidor
- Páginas completas
- Protocolo HTTP
 - HTTP GET: solicitar página
 - HTTP POST: enviar formulario



- Asynchronous JavaScript and XML
- Brevemente: uso de JavaScript para comunicarse de forma asincrónica con el servidor y traer trozos de páginas sin recargar y sin una nueva solicitud de página completa
- Características:
 - La lógica está todavía en el servidor, sólo una pequeña parte en el cliente (navegador)
 - La primera solicitud pide la página completa y las siguientes solicitudes piden partes
 - Protocolo HTTP
 - HTTP GET: solicitar página
 - HTTP POST: enviar formulario
 - XMLHttpRequest, intercambio de información asíncrono
 - Con la respuesta del servidor se modifica la página completa que trajimos en primer lugar, tal vez ocultando alguna parte, mostrando una nueva pieza, ...

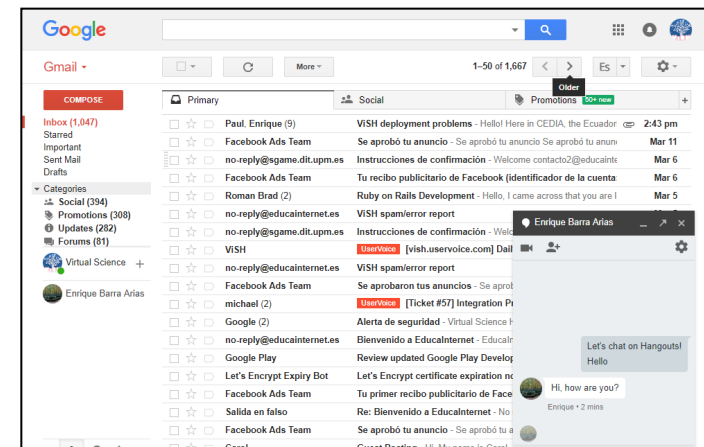
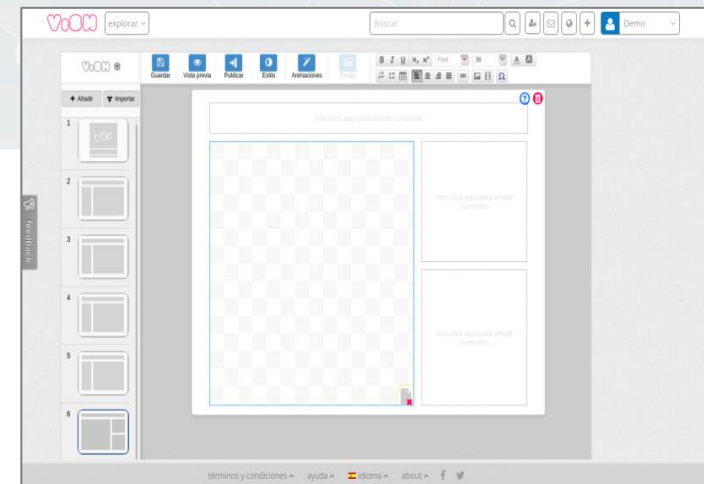
- Single-Page Application
- Arquitectura de las aplicaciones web
- Más similares a aplicaciones de desktop
- Ejemplo: Gmail, Google Maps, Drive, Trello...

➤ Características

- La mayor parte de la lógica está en el cliente (el servidor es sólo para almacenar datos y no mucho más)
- Con la primera solicitud el navegador trae la aplicación web
- Posteriormente se utilizan otras tecnologías para comunicarse con el servidor y enviar y recibir datos
 - AJAX
 - Websockets

➤ Idea importante:

- Manipular el DOM es muy costoso computacionalmente





Vue.js



ANGULAR

