



ARQUITECTURA DE LAS APLICACIONES WEB

© 2017, ACTIBYTI PROJECT SLU, Barcelona
Autor: Ricardo Ahumada

ÍNDICE DE CONTENIDOS

1. Caso práctico
2. Visión global de aplicaciones web
3. Arquitectura de las aplicaciones web
4. El Protocolo HTTP

1

CASO PRÁCTICO



Caso Práctico: Banana Apps

Banana Apps diseña e implementa aplicaciones de gestión no triviales multicanal para empresas.

Actualmente, el equipo técnico de Banana está considerando el desarrollo de una Red Social basada en videos “**BananaTube**”, para su cliente preferente; la idea es que BananaTube sea el próximo boom! de las redes sociales; permitiendo a sus usuarios gestionar videos, exponerlos en un muro, comentarlos, calificarlos y compartirlos en varios canales; además de esto, BananaTube expondrá una API, para poder integrarse con otros sistemas y así apalancar el desarrollo.

Al igual que con todos sus productos, tienen la intención de diseñar una Aplicación que cumpla con las tendencias actuales y asegurar la calidad en la implementación.

Banana Apps entrega productos llave en mano. Para este proyecto el equipo se tendrá que ocupar de todos los aspectos del desarrollo del frontend y una API REST, así com la puesta en producción del mismo.



Discutamos sobre lo que ya conocemos y lo que necesitamos

- Qué son aplicaciones no triviales-multicanal
- Porqué tecnologías web?
- Qué implica construir una aplicación web
- Qué Decisiones vamos a tener que adoptar?
- Qué vamos a necesitar?
- Estaremos solos?

2

Visión global de aplicaciones web

Aplicaciones web

- Son aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un **navegador**.
- Es una aplicación software compuesto por páginas web que se codifican en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.



Aplicación web

- Existen aplicaciones como los webmails, wikis, weblogs, tiendas online... son ejemplos bien conocidos de aplicaciones web.
- Otras aplicaciones son más del ámbito empresarial como ERPs, CMSs, CRMs, aplicaciones de gestión
- Una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información.
- Por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

Tipos de aplicación web

Existen 2 tipos de aplicaciones Web:

➤ Orientada a la presentación

- Genera paginas web interactivas que contienen varios tipos de lenguaje de marca (HTML, XML, etc.) y contenido dinámico en respuesta a peticiones.



➤ Orientada al servicio

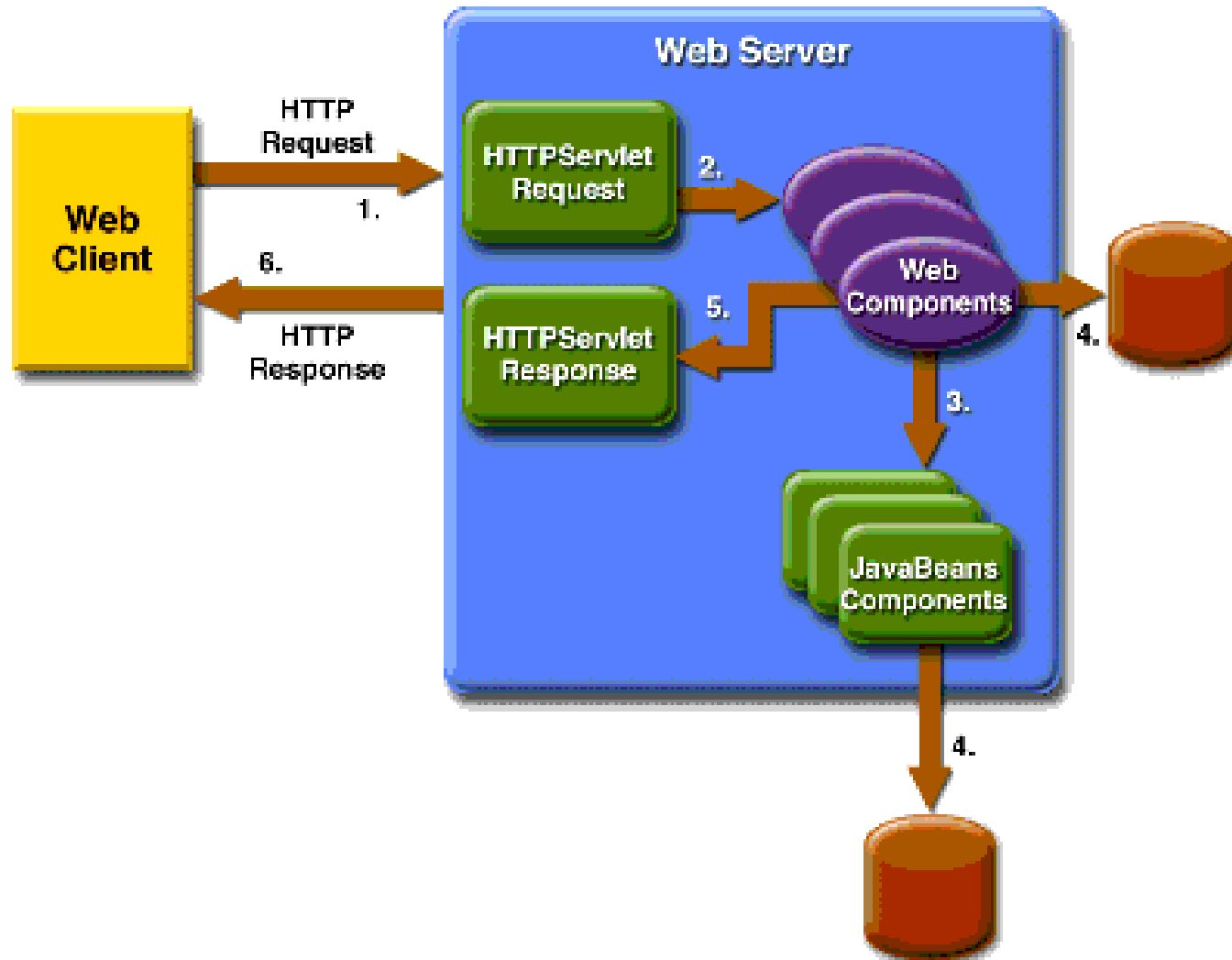
- Estas paginas implementan el punto final del servicio web.
- Son los servicios Web
- Las aplicaciones orientadas a la presentación frecuentemente son clientes de las aplicaciones web orientadas al servicio.



Interacción entre un cliente y una aplicación Web.

- El cliente envía una petición HTTP al servidor web
- Las tecnologías del servidor convierten la petición en un objeto que representa la petición
- Esta petición es entregada a un componente Web, el cual puede interactuar con otros componentes para generar un contenido dinámico
- El componente web puede generar un objeto que representa la respuesta
- El servidor web convierte este objeto en una respuesta HTTP y es enviada a su cliente.

Interacción – Ejemplo Java Server





Pongámoslo en práctica

- Examina la aplicación <https://freedcamp.com/>
 - › Date de alta
 - › Crea proyectos
 - › Observa sus funcionalidades
 - › Qué valor que aporta?

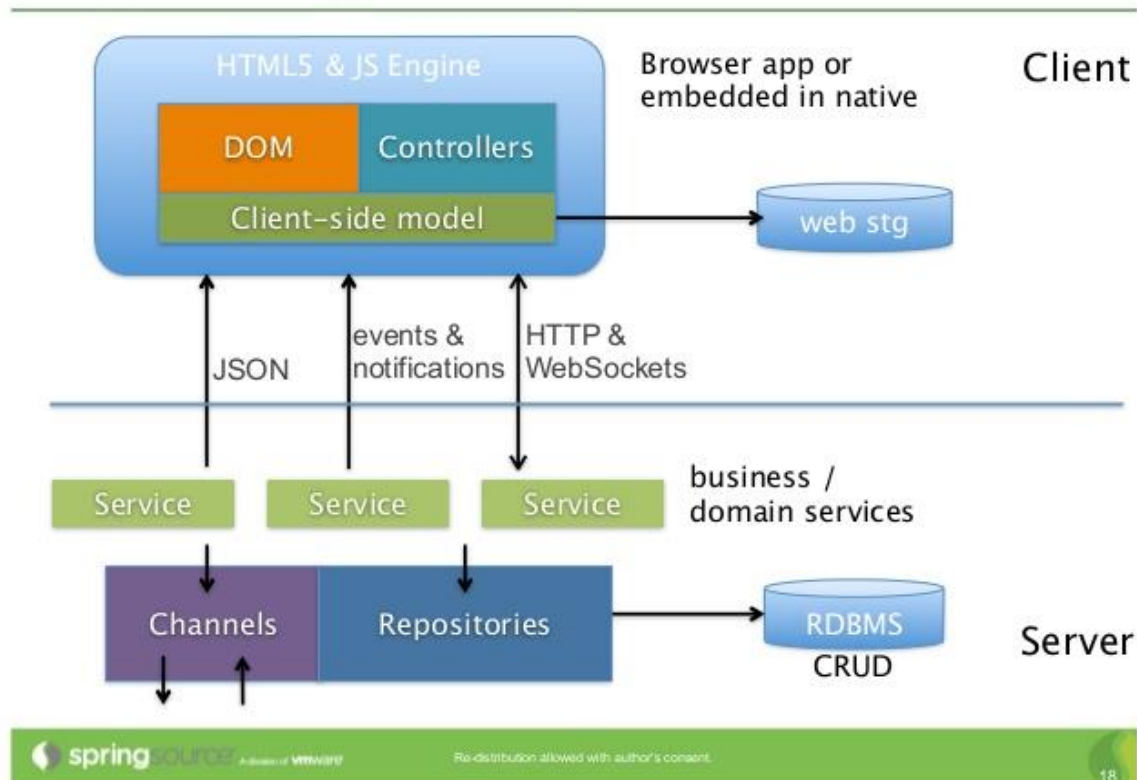
3

Arquitectura de las aplicaciones web

Arquitectura

- Una Arquitectura define un conjunto de elementos, conectores, restricciones y un sistema de control que caracterizan a un sistema o a una familia de sistemas.

From server-side app to smart clients and services



Separación cliente - servidor

- La separación entre cliente y servidor es una **separación de tipo lógico**, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.
- Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc.
- Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma

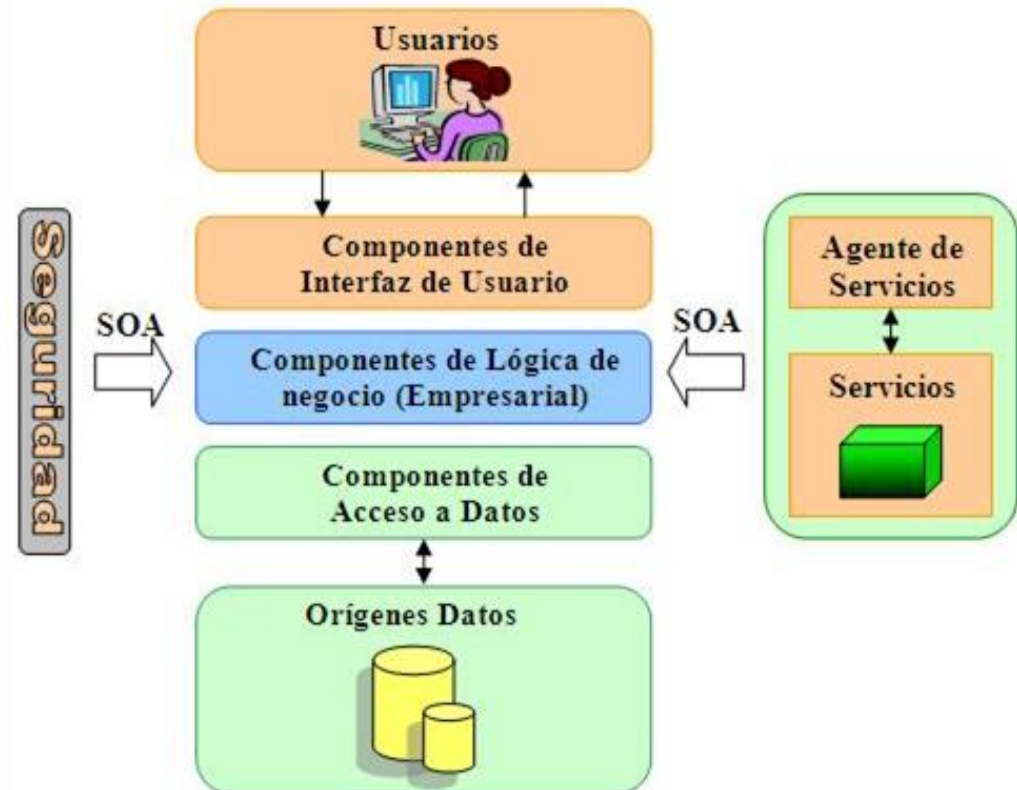
Red cliente-servidor

- Red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados.
- Todas las gestiones que se realizan se concentran en el servidor. En él se disponen las peticiones provenientes de los clientes; los archivos que son de uso público y de uso restringido; los archivos de sólo lectura y los que pueden ser modificados, etc.



Arquitectura en 3 capas

- Consiste en separar en el desarrollo de un producto todas las capas lógicas identificables.
 - Capa de datos
 - Capa de negocios
 - Capa de presentación
- Lo ideal es que cada capa atienda a una responsabilidad concreta y lo más aislada posible.



¿Qué hace cada capa?

- **La capa de presentación:** Es la que se visualiza el cliente y es con la que interactúa.
 - La interface web
 - Generación de interface web (página web)

- **La capa negocio:** Esta incluye la lógica del negocio, en ella se hacen los procesos, es una capa intermedia entre presentación y datos y sirve como puente entre estas dos capas.
 - Reglas del negocios
 - Validaciones
 - Cálculos
 - Flujos y procesos

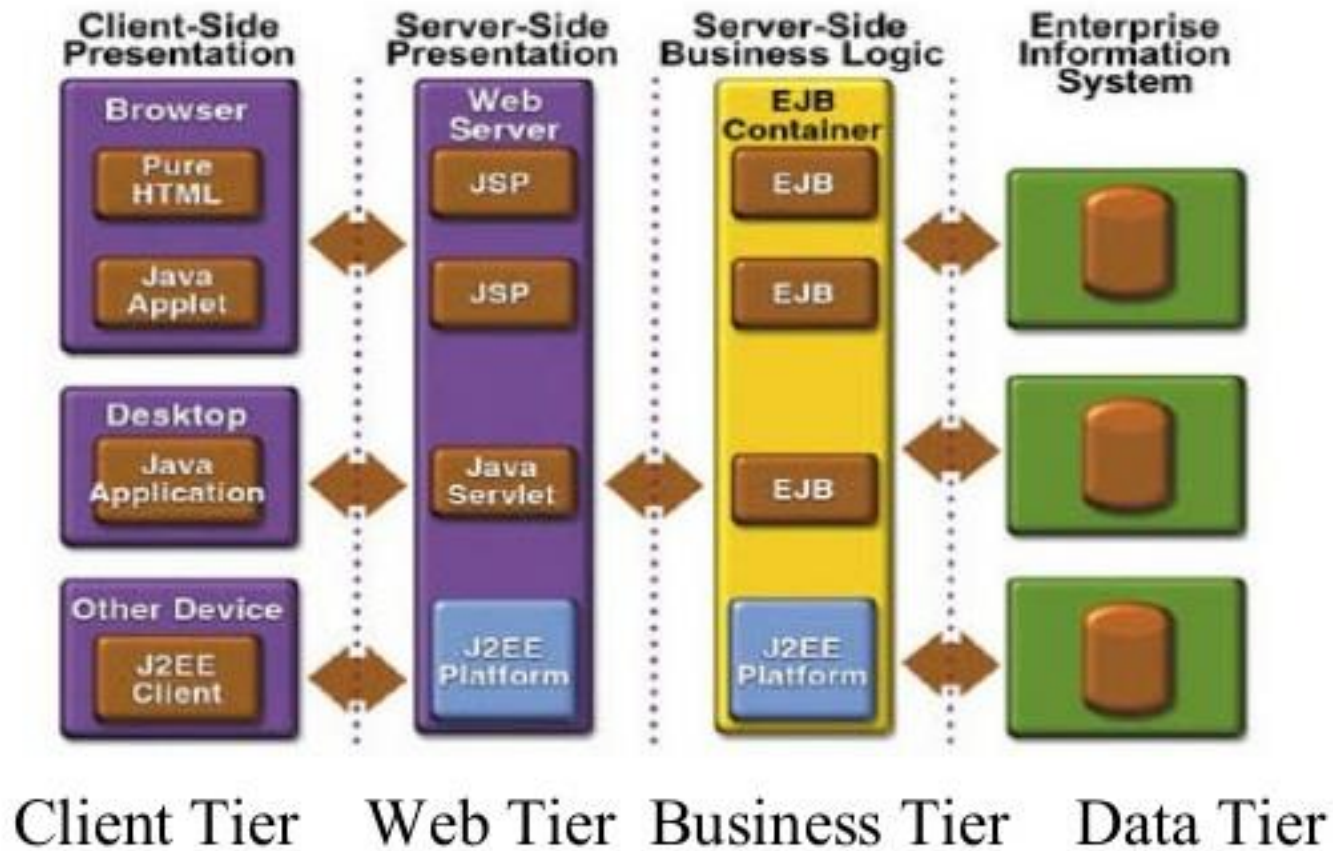
- **La capa de datos:** Se encarga de administrar el registro de datos (almacenar y recuperar)
 - Base de datos
 - Tablas
 - Procedimientos almacenados
 - Componentes de datos

La Web

Al conectarnos a internet estamos navegando en 3 capas.

- Al abrir un formulario web de inscripción (capa de presentación)
- Después de enviar la información esta es verificada (capa de negocios).
- Finalmente la información es grabada en una base de datos (capa de datos).

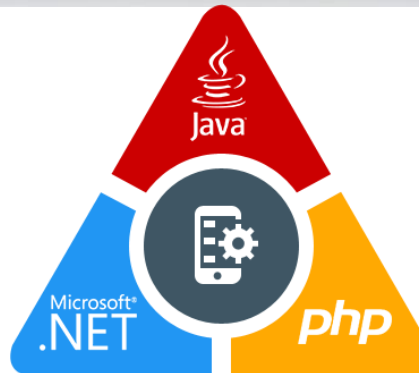
Ejemplo –Arquitectura Capas App Java



Tecnologías de las aplicaciones web



Open Web Technologies and You



Aplicación web JEE



Open Web Technologies and You



LENGUAJES DE PROGRAMACION DEL LADO DEL CLIENTE



Lenguajes del lado del cliente

- Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio.
- Para que la pagina se pueda ver bien es necesario tener instalados los plug-in adecuados.
- El navegador es una especie de aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden.
- Los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

HTML

- HTML indica al navegador donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página.
- Sirve para crear la estructura de la página
- Este lenguaje consta de etiquetas que tienen esta forma <div> o <P>. Cada etiqueta significa una cosa.

JAVASCRIPT

- Javascript es un lenguaje de programación utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web.
- Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento.
- Hasta hace poco su uso se basaba en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.
- Ahora permite crear aplicaciones completas en el lado del cliente como las Aplicaciones de página única (SPA)

CSS

- CSS, es una tecnología que nos permite dar diseño a los componentes de una páginas web.

APPLETS DE JAVA

- Se trata de pequeños programas hechos en Java, que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página.
- Los applets de Java están programados en Java y precompilados.

FLASH

- Flash es una tecnología, y un programa, para crear efectos especiales en páginas web. Con Flash se hacen páginas dinámicas del lado del cliente.

LENGUAJES DE PROGRAMACION DEL LADO DEL SERVIDOR



Lenguajes del lado del servidor

- Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente.
- Los componentes que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente
- Son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

CGI

- Es el sistema más antiguo que existe para la programación de las páginas dinámicas de servidor.
- Actualmente se encuentra un poco desfasado por la dificultad con la que se desarrollan los programas y la pesada carga que supone para el servidor que los ejecuta.
- Los CGI se escriben habitualmente en el lenguaje Perl, C, C++ o Visual Basic.

PEARL

- Perl es un lenguaje de programación interpretado.
- Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito.

PHP

- PHP es el acrónimo de Hipertext Preprocesor.
- Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

ASP.net

- ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor.
- ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).

JSP, Servlets

- Es una tecnología orientada a crear páginas web con programación en Java.
- Con JSP se hacen aplicaciones web que se ejecutan en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, JSP se escriben con editores HTML/XML habituales.

Tecnologías por capa

➤ **Presentación**

- HTML, CSS, JS
- Flash, Applets

➤ **Negocio**

- Lenguajes de Componentes (Servlets, JSP, Java Beans, EJBs, c#, vb, c++, j#)
- Componentes Locales
- Componentes Web (Servicios web)
- Comunicación entre componentes utilizando el SOAP y XML

➤ **Datos**

- ADO .NET
- SQL SERVER 20000, Oracle, Postgres, Mysql
- Procedimientos almacenados
- Componentes (Java Beans, c#, vb, c++, j#)
- XML

4

El Protocolo HTTP

HTTP: Hypertext Transfer Protocol

- Es un conjunto de reglas que rigen la transferencia de datos en una comunicación Web.
- El protocolo más usado de Internet.

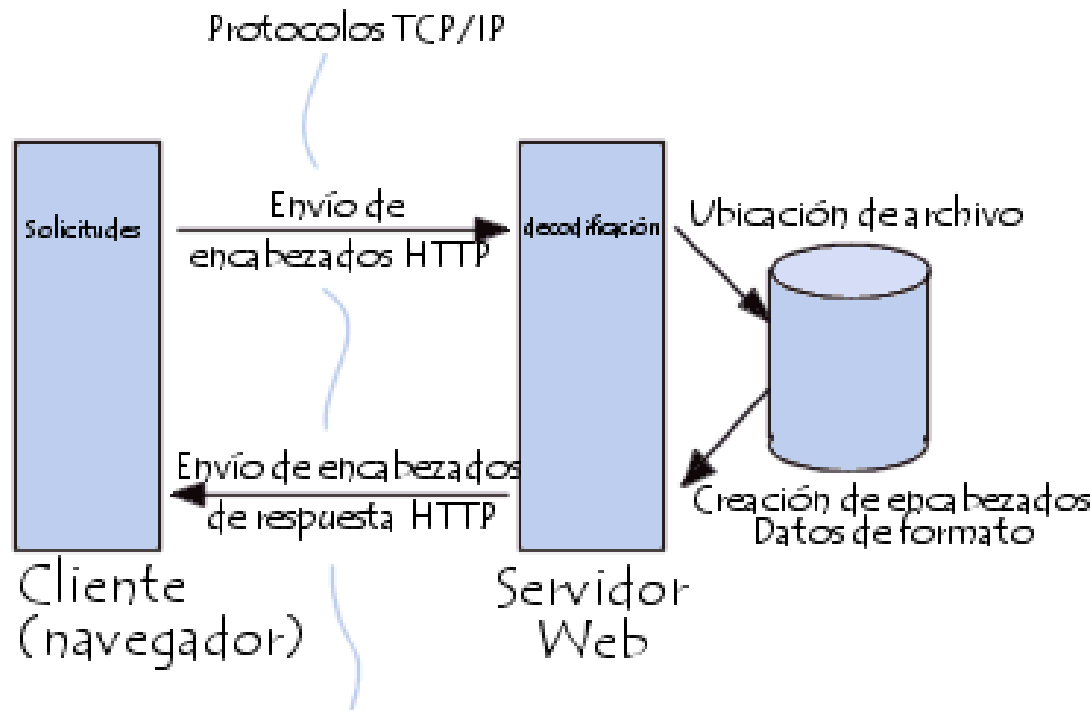
Su propósito:

- Transferencia de archivos entre un cliente (el navegador) y un servidor web.
- Se usa tanto para que el navegador pida una pagina a un servidor como para que este envíe la pagina solicitada al navegador.
- Esta basado en el envío de comandos y respuestas de texto en formato ASCII.



Comunicación cliente - servidor

- **Conexión:** establecimiento de una conexión cliente-servidor. El puerto TCP/IP 80 es el más conocido.
- **Solicitud:** envío del cliente de un mensaje de solicitud al servidor.
- **Respuesta:** envío del servidor de una respuesta al cliente.
- **Cierre:** fin de la conexión por parte del cliente y el servidor.



Solicitud HTTP (request)

- Conjunto de líneas de texto que el cliente envía al servidor
- **Línea de solicitud**
 - 3 elementos separados por un espacio
comando dirección URL versión del protocolo
 - Protocolo: HTTP/1.0 o HTTP/1.1
- **Campos de encabezado**
 - Líneas con información sobre la solicitud y el cliente (navegador, sistema operativo, etc.)
 - En cada línea:
tipo de encabezado: valor del encabezado.
- **Cuerpo de la solicitud**
 - Permite envío de datos
 - Separada por una línea en blanco de las precedentes

Esquema de solicitud

MÉTODO VERSIÓN URL
ENCABEZADO: Valor
... ENCABEZADO: Valor

CUERPO DE LA SOLICITUD

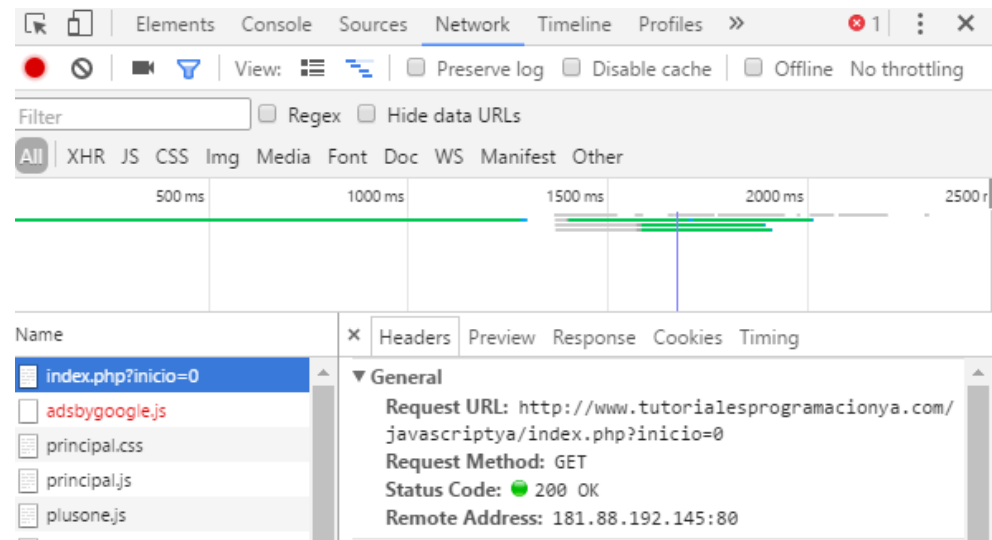
Ejemplo

```
GET http://es.kioskea.net HTTP/1.0 Accept : Text/html If-Modified-Since : Saturday, 15-  
January-2000 14:37:11 GMT User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)
```



Pongámoslo en práctica - Observando requests

- Accede a una página web
- Activa el panel de desarrollador de tu navegador (tecla F12)
- Accede a la pestaña Network
- Haz clic sobre una de las peticiones
- Analiza las peticiones, respuesta y timing de la petición



Comandos o métodos HTTP

Comando	Descripción
GET	Solicita el recurso ubicado en la URL especificada
HEAD	Solicita el encabezado del recurso ubicado en la URL especificada
POST	Envía datos al programa ubicado en la URL especificada
PUT	Envía datos a la URL especificada
DELETE	Borra el recurso ubicado en la URL especificada

Encabezados más comunes

Encabezado	Descripción
Accept	Tipo de contenido aceptado por el navegador (por ejemplo, texto/html). Consulte Tipos de MIME
Accept-Encoding	Codificación de datos que el navegador acepta
Authorization	Identificación del navegador en el servidor
Content-Encoding	Tipo de codificación para el cuerpo de la solicitud
Content-Type	Tipo de contenido del cuerpo de la solicitud (por ejemplo, texto/html). Consulte Tipos de MIME
Date	Fecha en que comienza la transferencia de datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
Referer	Dirección URL desde la cual se realizó la solicitud
User-Agent	Cadena con información sobre el cliente, por ejemplo, el nombre y la versión del navegador y el sistema operativo

Respuesta HTTP (response)

- Conjunto de líneas que el servidor envía al cliente
- **Línea de estado**
 - 3 elementos separados por un espacio:
versión del protocolo código de estado significado de código
 - Protocolo: HTTP/1.0 o HTTP/1.1
- **Campos de encabezado de respuesta**
tipo de encabezado: valor del encabezado.
- **Cuerpo de la respuesta**
 - Permite envío de datos desde el servidor
 - Separada por una línea en blanco de las precedentes

Esquema de respuesta

VERSIÓN-HTTP CÓDIGO EXPLICACIÓN
ENCABEZADO: Valor
... ENCABEZADO: Valor

CUERPO DE LA RESPUESTA

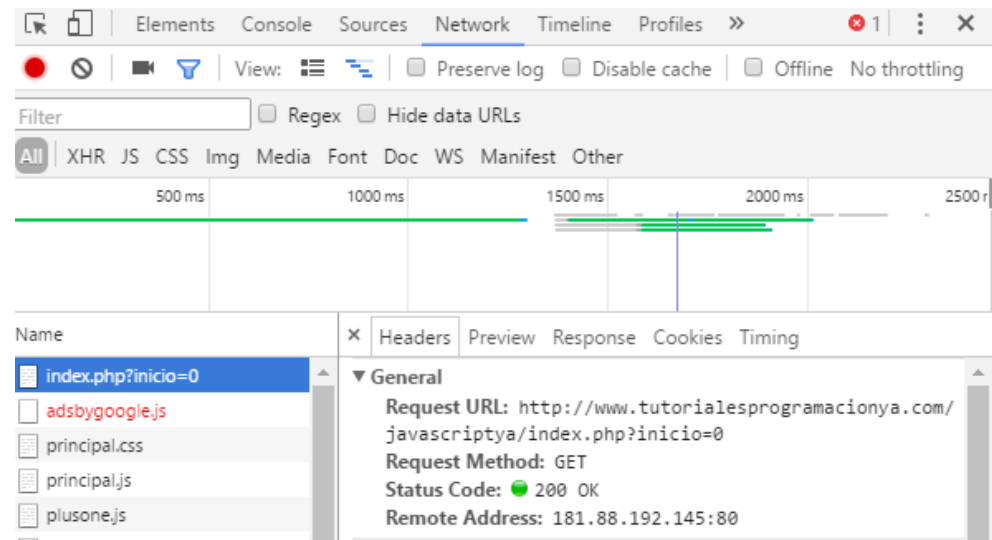
Ejemplo

```
HTTP/1.0 200 OK Date: Sat, 15 Jan 2000 14:37:12 GMT Server :  
Microsoft-IIS/2.0 Content-Type : text/HTML Content-Length : 1245  
Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT
```



Pongámoslo en práctica - Observando responses

- Accede a una página web
- Activa el panel de desarrollador de tu navegador (tecla F12)
- Accede a la pestaña Network
- Observa los datos de status
- Haz clic sobre una de las peticiones
- Analiza las respuestas



Encabezados de Respuesta

Nombre del encabezado	Descripción
Content-Encoding	Tipo de codificación para el cuerpo de la respuesta
Content-Language	Tipo de idioma en el cuerpo de la respuesta
Content-Length	Extensión del cuerpo de la respuesta
Content-Type	Tipo de contenido del cuerpo de la respuesta (por ejemplo, texto/html). Consulte Tipos de MIME
Date	Fecha en que comienza la transferencia de datos
Expires	Fecha límite de uso de los datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
Location	Redireccionamiento a una nueva dirección URL asociada con el documento
Server	Características del servidor que envió la respuesta

Códigos de Respuesta- Esquema

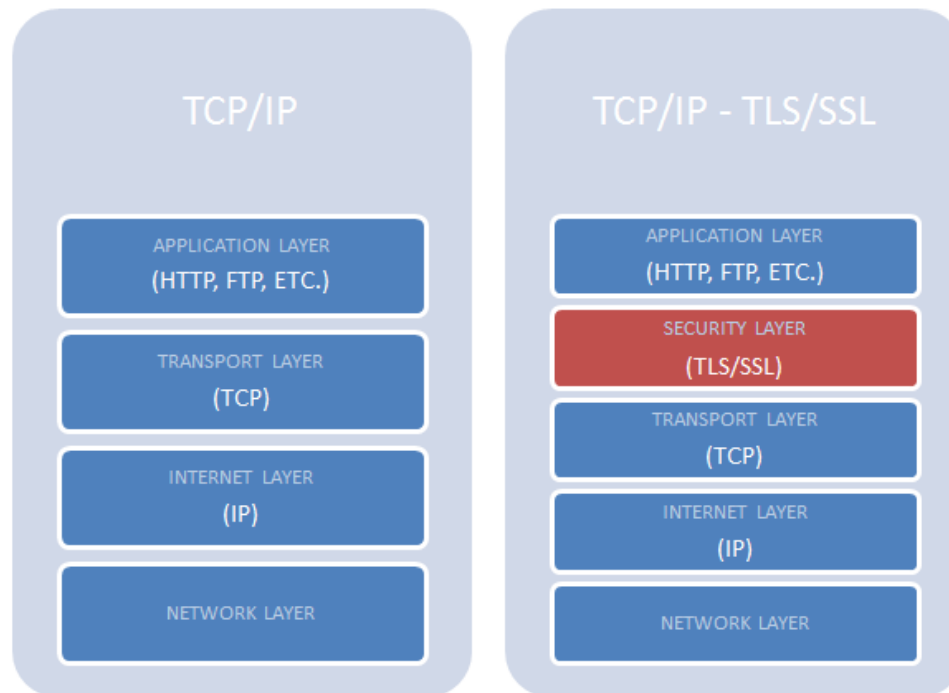
Código	Mensaje	Descripción
20x	Éxito	Estos códigos indican la correcta ejecución de la transacción
30x	Redirección	Estos códigos indican que el recurso ya no se encuentra en la ubicación especificada
40x	Error debido al cliente	Estos códigos indican que la solicitud es incorrecta
50x	Error debido al servidor	Estos códigos indican que existe un error interno en el servidor

Códigos de Respuesta más comunes

Código	Mensaje	Descripción
202	ACCEPTED	La solicitud ha sido aceptada, pero el procedimiento que sigue no se ha llevado a cabo
204	NO RESPONSE	El servidor ha recibido la solicitud, pero no hay información de respuesta
400	BAD REQUEST	La sintaxis de la solicitud se encuentra formulada de manera errónea o es imposible de responder
401	UNAUTHORIZED	Los parámetros del mensaje aportan las especificaciones de formularios de autorización que se admiten. El cliente debe reformular la solicitud con los datos de autorización correctos
403	FORBIDDEN	El acceso al recurso simplemente se deniega
404	NOT FOUND	Un clásico. El servidor no halló nada en la dirección especificada. Se ha abandonado sin dejar una dirección para redireccionar... :)
500	INTERNAL ERROR	El servidor encontró una condición inesperada que le impide seguir con la solicitud (una de esas cosas que les suceden a los servidores...)

HTTPS: Hypertext Transfer Protocol Secure

- HTTP que viaja sobre SSL/TLS (Secure Sockets Layer/Transport Layer Security)



HTTPS: Proceso



- Clave pública (para encriptar) y privada (para des-encriptar)
- El certificado SSL: contienen clave pública y dominios en que se pueden usar
- Las CA(Certificate Authority) emiten SSL sobre dominio a su propietario

Encriptación

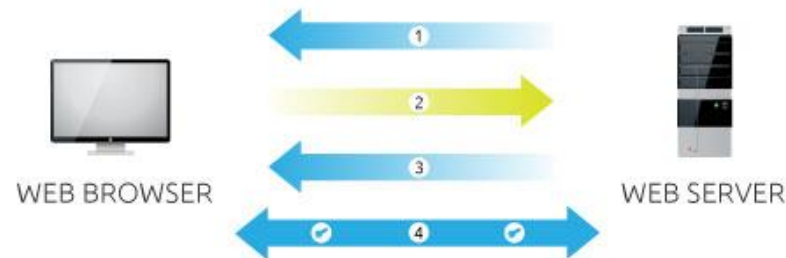
- Encriptación asimétrica
 - › public-key cryptography
 - › Usa 2 claves: Encriptar (pública) - Desencriptar (privada)
 - › Algoritmo RSA - 1024 or 2048 bits 14 billiones de años desencriptar
- Encriptación simétrica
 - › Encriptación pre-compartida
 - › Una sola clave para encriptar-desencriptar
 - › 128-256 bits → fácil de romper
- SSL no dicta qué encriptación usar
- Se escoge en función de la carga computacional y facilidad de distribución

Cómo se usa la encriptación

➤ PKI: Public Key Infrastructure

- hardware, software, gente, políticas y procedimientos necesarios para crear, gestionar, distribuir, usar, almacenar y revocar certificados digitales
- PKI enlaza las claves con identidades de usuarios a través de un CA (Certificate Authority)
- Usan cripto-sistema híbrido.

➤ Proceso



- El servidor envía clave asimétrica
- El navegador crea clave simétrica (de sesión) y la envía encriptada con la clave asimétrica del servidor.
- El navegador descripta la clave simétrica (de sesión).
- Servidor y navegador usan la clave simétrica (sólo para la sesión)

Encriptación de claves

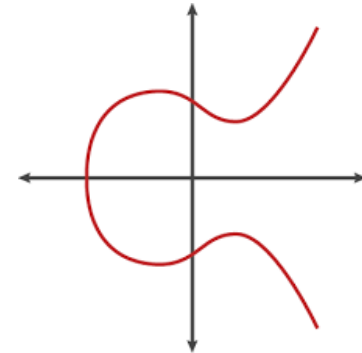
Claves públicas

➤ RSA (Rivest-Shamir-Adleman)

- › Usa la dificultad de factorizar números enteros grandes.
- › Se crea en función de 2 números primos grandes, junto a un valor auxiliar

➤ ECC (Elliptic curve cryptography)

- › Basada en la estructura de curvas elípticas
- › Asume que descubrir el logaritmo discreto de una curva elíptica aleatoria en conexión con una base pública conocida es impracticable
- › La clave puede ser más pequeña que la de RSA (más velocidad y seguridad)
- › No soportada por todas las aplicaciones y servicios



Claves Pre compartidas

- › Wofish, AES, or Blowfish. (AES la más popular)
- › Stream Cipher: aplica encriptación a cada bit
- › Block Cipher: aplica encriptación a cada bloque (64 bits) [más común]



Pongámoslo en práctica

- Vamos a realizar una auditoría de HTTPS
- Para ello accede al sitio <https://www.ssllabs.com/ssltest/>
- Y prueba algunas páginas conocidas.



 **netmind**

WeKnowIT

Barcelona

C. Almogàvers, 123
08018 Barcelona
Tel. 93 304.17.20
Fax. 93 304.17.22

Madrid

Plaza Carlos Trias Bertrán, 7
28020 Madrid
Tel. 91 442.77.03
Fax. 91 442.77.07

www.netmind.es