
Introducción al desarrollo de aplicaciones Web

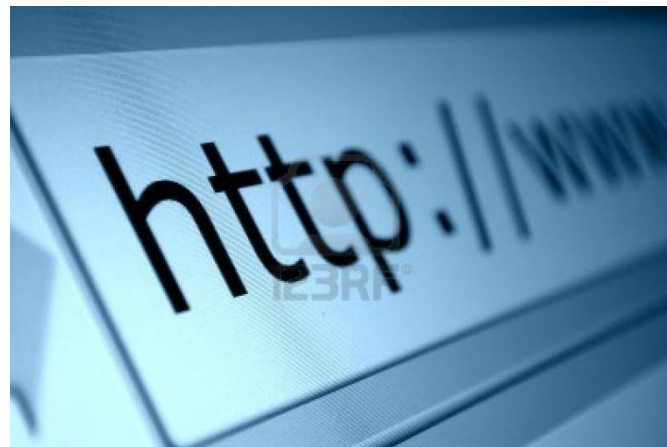
Abraham Sánchez López
FCC/BUAP
Grupo MOVIS

Introducción

- El diseño y desarrollo de aplicaciones Web consiste en implementar las necesidades, objetivos o ideas de los clientes en la Internet utilizando las tecnologías más idóneas según la naturaleza del proyecto.
- Las aplicaciones Web ofrecen servicios a los usuarios de Internet que acceden utilizando un navegador.
- Las aplicaciones Web pueden ser de acceso público como tiendas virtuales, periódicos digitales, portales de Internet.
- Las aplicaciones Web de acceso restringido como son las intranets, mejoran las gestiones internas de una empresa como el reporte de horas de su personal, gestión de proyectos y tareas, control de presencia, gestores documentales.
- O incluso, el uso de extranets aumenta y mejora el servicio con sus distribuidores, clientes, proveedores, comerciales y colaboradores externos.
- Cómo desarrollar una aplicación Web? Tecnologías Web + Ingeniería Web

Fundamentos de la Web

- El éxito espectacular de la Web se basa en dos aspectos fundamentales:
 - El protocolo HTTP y el lenguaje HTML.
- El primero permite una implementación simple y sencilla de un sistema de comunicaciones que nos permite enviar cualquier tipo de archivos de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que los servidores con poco “performance” atiendan miles de peticiones y reduzcan los costos de despliegue.
- El otro nos proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple.



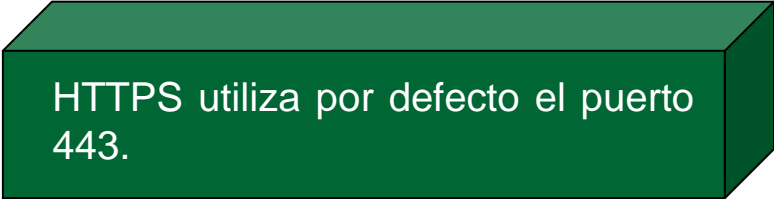
El protocolo HTTP, I

- El protocolo HTTP (*HyperText Transfer Protocol*) es el protocolo base de la WWW.
- Se trata de un protocolo simple, orientado a conexión y sin estado.
- La razón de que esté orientado a conexión es que emplea para su funcionamiento un protocolo de comunicaciones (TCP, *Transport Control Protocol*) de modo conectado, un protocolo que establece un canal de comunicaciones de extremo a extremo (entre el cliente y el servidor) por el que pasa el flujo de bytes que constituyen los datos que hay que transferir, en contraposición a los protocolos de datagrama o no orientados a conexión que dividen los datos en pequeños paquetes (datagramas) y los envían, pudiendo llegar por diferentes vías del servidor al cliente.

HTTP utiliza el puerto 80 (equivalente de alguna forma al identificador de conexión o de servicio TCP) para todas las conexiones por defecto (podemos utilizar otros puertos diferentes del 80).

El protocolo HTTP, II

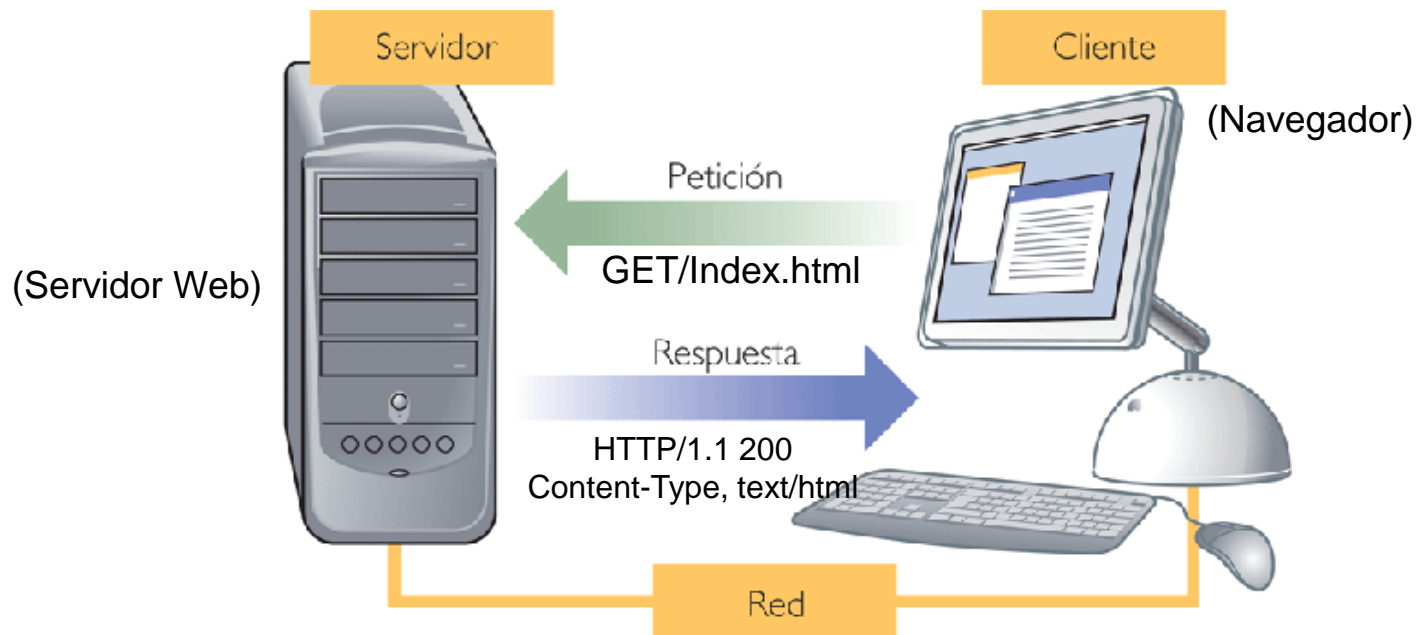
- El protocolo no mantiene estado, es decir, cada transferencia de datos es una conexión independiente de la anterior, sin relación alguna entre ellas, hasta el punto de que para transferir una página Web tenemos que enviar el código HTML del texto, así como las imágenes que la componen, pues en la especificación inicial de HTTP, la 1.0, se abrían y usaban tantas conexiones como componentes tenía la página, transfiriéndose por cada conexión un componente (el texto de la página o cada una de las imágenes).
- Existe una variante de HTTP llamada HTTPS (S por *secure*) que utiliza el protocolo de seguridad SSL (*Secure sockets Layer*) para cifrar y autenticar el tráfico entre cliente y servidor, siendo ésta muy usada por los servidores Web de comercio electrónico, así como por aquellos que contienen información personal o confidencial.



HTTPS utiliza por defecto el puerto 443.

El protocolo HTTP, III

- De manera esquemática, el funcionamiento de HTTP es el siguiente: el cliente establece una conexión TCP hacia el servidor, hacia el puerto HTTP (o el indicado en la dirección de conexión), envía un comando HTTP de petición de un recurso (junto con algunas cabeceras informativas) y por la misma conexión el servidor responde con los datos solicitados y con algunas cabeceras informativas.



El lenguaje HTML

- La otra parte del éxito de la WWW ha sido el lenguaje HTML (*HyperText Markup Language*).
- *Se trata de un lenguaje de marcas (se utiliza insertando marcas en el interior del texto) que nos permite representar de forma rica el contenido y también referenciar otros recursos (imágenes, etc.), enlaces a otros documentos (la característica más destacada de la WWW), mostrar formularios para posteriormente procesarlos, etc.*
- El lenguaje HTML actualmente se encuentra en la versión 5 y este proporciona funcionalidades más avanzadas para crear páginas más ricas en contenido.
- Además se ha definido una especificación compatible con HTML, el XHTML (*eXtensible HyperText Markup Language*) que se suele definir como una versión XML validable de HTML, proporcionándonos un XML Schema que nos permite validar el documento para comprobar si está bien formado, etc.

Desventajas de las aplicaciones Web

- Las arquitecturas y técnicas en programación Web buscan:
 - Máxima compatibilidad con los navegadores (clientes potenciales).
 - Eficiencia del lado del servidor (soportar el máximo número de conexiones concurrentes).
 - Creación de contenidos llamativos y dinámicos sin sacrificar los dos puntos anteriores.
- De hecho, HTML falla en este último punto!!
- Se necesitan lenguajes de programación más potentes que permitan “mini” aplicaciones dentro de la Web, por ejemplo:
 - Carritos de compras (aplicaciones de comercio electrónico).
 - Contador de visitas.
 - Web mail.
 - Registro y entrada de un sistema.
 - Transferencias y operaciones bancarias.
 - Etc.



Ejemplos de aplicaciones Web

WebMail Ingreso

NOMBRE DE USUARIO:

CONTRASEÑA:

Remember Name & Password

INGRESO

Avuda

STIHL

Login

Ingrese su código de usuario y contraseña.

Usuario:

Contraseña:

Login

http://caixa/tyck/

"la Caixa" [Accesibilidad](#) [Atenci](#)

Particulares | [Empresas](#) | [LIXA](#) | [Ver más \(Mayores, Extranjeros...\)](#)

Línea Abierta [Otras idiomas](#)

Identificación:

Nº secreto personal PIN1:

Clave:

Entrar

[Reservar ahora](#) [Ayuda](#)

[Datos de alta - Demostración](#)
[Usuario Línea Abierta Windows](#)
[Otras formas de conectarse](#)

Libreta Vivienda

Te ayudamos a que la compra de tu piso resulte mucho más fácil

Aprovecha sus ventajas fiscales antes de fin de año

Ahorra para tu piso y aprovecha las ventajas fiscales

Depósito 5% TAE a 1 año

Contratando el plan de pensiones [PlanCaixa Triple 5](#).

Gratis, 6 € en llamadas

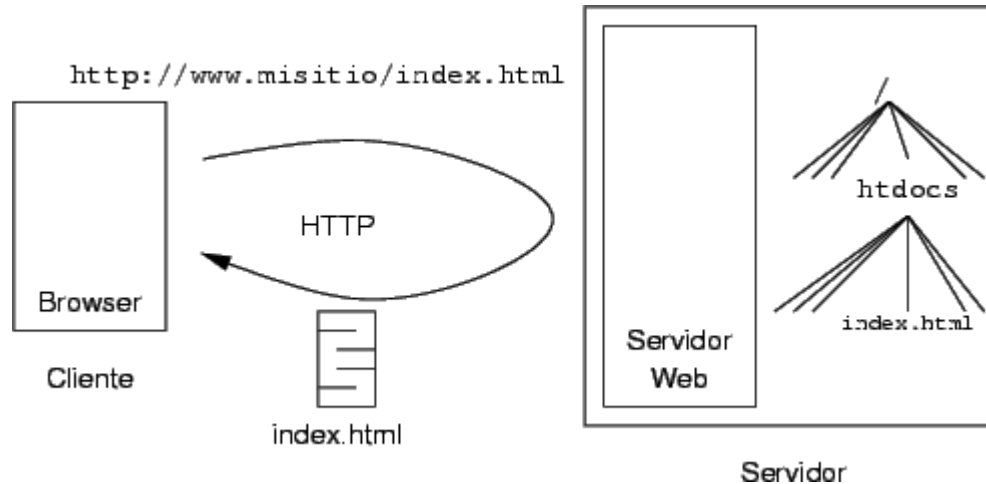
Te regalamos 6 € en llamadas por cada [CaixaGiro](#) realizado.

Productos y servicios

- ▶ Bolsa Abierta
- ▶ Puntos Estrella
- ▶ Tarjetas
- ▶ Préstamos
- ▶ Ahorro

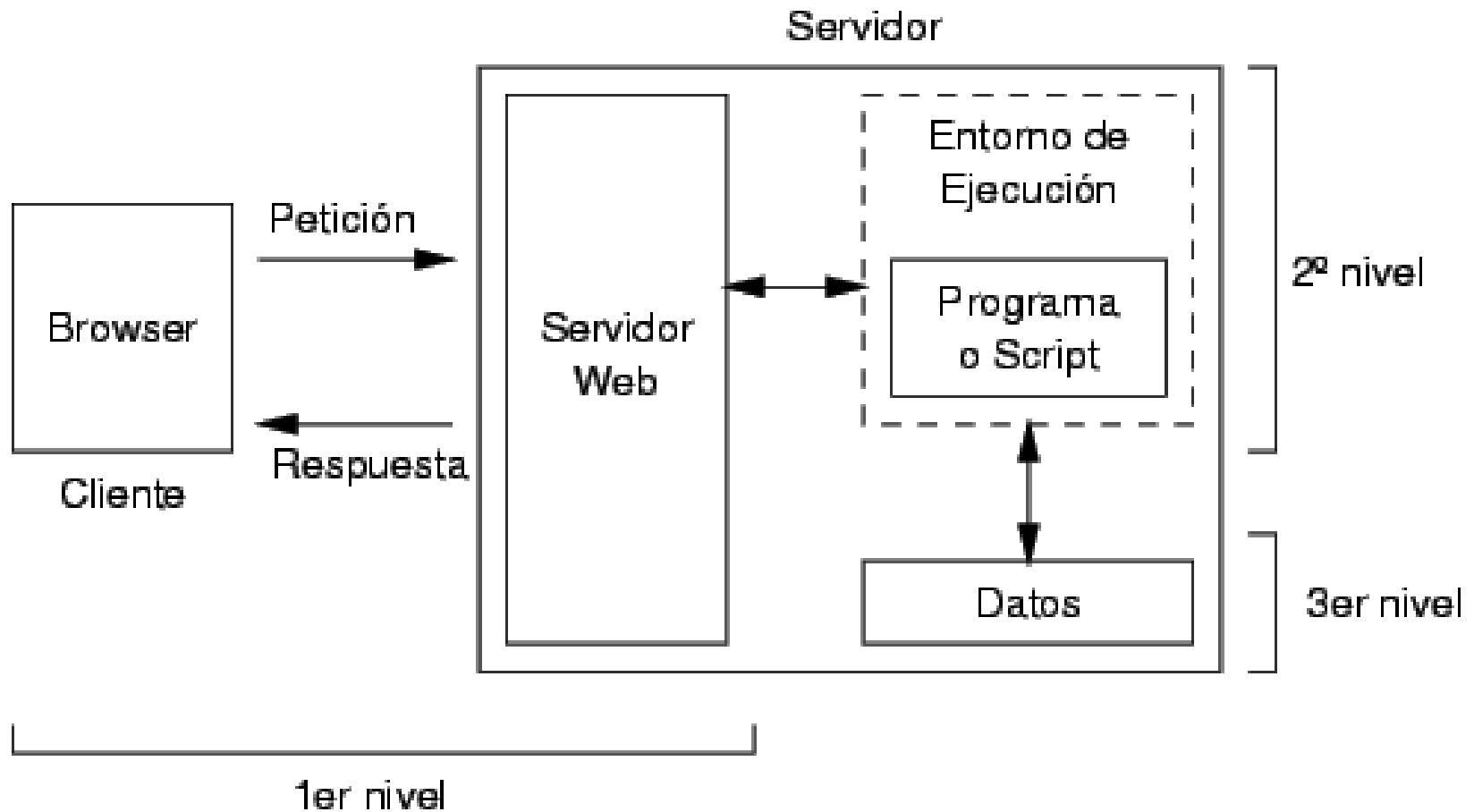
Arquitectura de las aplicaciones Web

- **Arquitectura de dos niveles:** es la más simple, se tiene el nivel del “cliente” y el nivel del “servidor”.



- **Arquitectura de tres niveles:** El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel se refiere habitualmente a algún tipo de programa o *script*. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Arquitectura de tres niveles



Lenguajes de programación del lado del cliente

- Los programas del lado del cliente están incluidos dentro de la página HTML, se descargan del servidor junto con este.
- Los programas se ejecutan dentro del ámbito del browser.
- Tecnologías y lenguajes del lado del cliente:
 - * Navegadores para la Web
 - * HTML
 - * Javascript y Vbscript
 - * Applets en Java
 - * Flash (lenguaje ActionScript)
 - * XML
 - * PDF
 - * AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML).



Algunos navegadores Web

- Amaya
- Epiphany
- Galeon
- Internet Explorer
- Konqueror
- Lynx
- Mozilla suite navigator
- Mozilla Firefox
- Netscape Navigator
- Opera
- Safari
- Shiira
- Maik Navigator
- Chrome



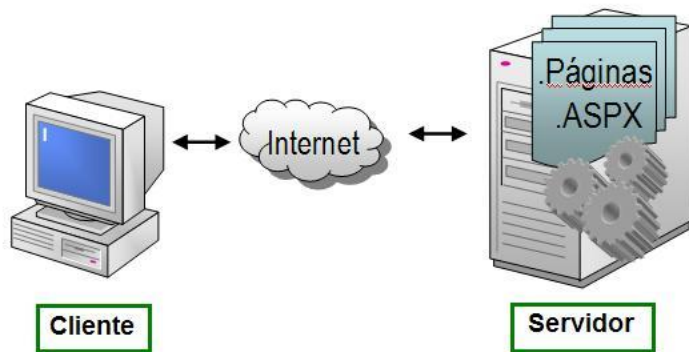
Tecnologías y lenguajes del lado del cliente

- Algunos de estos lenguajes y tecnologías requieren de un programa especial (plug-in) instalado en la computadora del usuario.
Ejemplo: Adobe Flash Player.
- Un **complemento** (o **plug-in** en inglés) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.
- Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

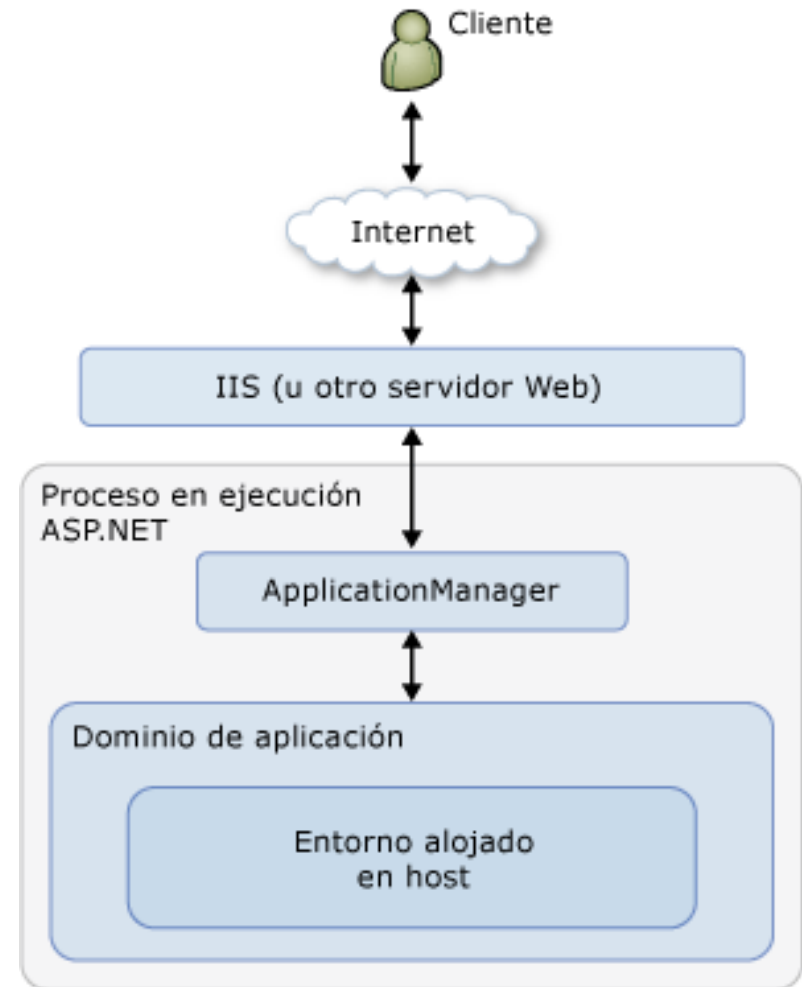


Lenguajes de programación del lado del servidor

- Se ejecutan en el servidor de Web y son dependientes de la plataforma del servidor.

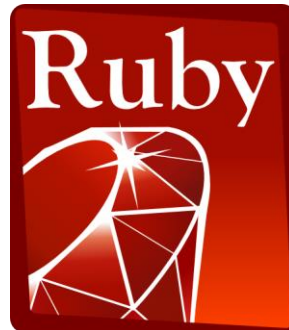


- Se usan para acceder a recursos del servidor, como bases de datos y generación de contenido dinámico para las páginas.
- Por ejemplo, el ámbito de ejecución de una página ASP.NET.



Algunos lenguajes del lado del servidor

- Algunos ejemplos de lenguajes del lado del servidor:
 - ASP, ASP.NET (son tecnologías que soportan diferentes lenguajes como VB, C#, C++, etc.).
 - PHP
 - JSP
 - Perl
 - Ruby
 - Python
 - XML



Servidor Web

- Un servidor Web es un programa que implementa el *protocolo HTTP* .
- Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que se llama hipertextos, páginas Web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.
- Algunos ejemplos:
 - ❑ CERN httpd
 - ❑ Apache (Libre, servidor más usado del mundo, según Wikipedia)
 - ❑ IIS
 - ❑ Resin
 - ❑ Tomcat (Libre, del proyecto Jakarta de Apache)
 - ❑ Geronimo (Libre, orientado a J2EE, del proyecto Jakarta de Apache, actualmente se encuentra en desarrollo) .
 - ❑ JBoss
 - ❑ JOnAS
 - ❑ Cherokee

Ambientes para desarrollo Web

- Los IDE (ambientes integrados de desarrollo) para aplicaciones Web son muy numerosos.
- Se deben considerar los que permitan trabajar con los diferentes lenguajes para la Web.
- Algunos son específicos para lenguajes del lado del servidor. Por ejemplo, Visual Studio solo soporta ASP.NET del lado del servidor.
- Existen IDE's libres y gratuitos de buena calidad.
- Algunos ejemplos de IDE's para la Web:
 - Microsoft Visual Studio.
 - Microsoft Web Developer Express.
 - Mono (para ASP.NET).
 - NetBeans.
 - Jbuilder.
 - Eclipse.



Metodologías para desarrollo Web, I

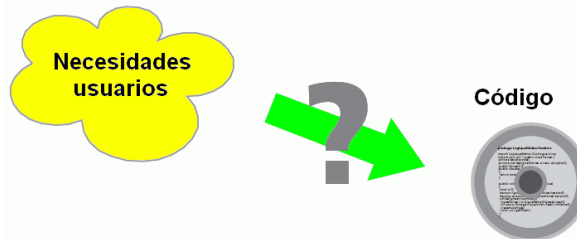
- Existen muchas propuestas para el desarrollo de aplicaciones Web, a continuación se resumen las más importantes:
 - * HDM (Hypermedia Design Model)
 - * RMM (Relationship Management Method)
 - * EORM (Enhanced Object Relationship Methodology)
 - * OOHDM (Object-Oriented Hypermedia Design Method)
 - * WSDM (Web Site Design Method)
 - * SOHDM (Scenario-based Object-oriented Hypermedia Design Methodology)
 - * RNA (Relationship-Navigational Analysis)
 - * HFPM (Hypermedia Flexible Process Modelling Strategy)
 - * Building Web Applications with UML (Conallen)
 - * WebML (Web Modelling Language)
 - * UWE (UML-based Web Engineering)
 - * W2000
 - * Proyecto UWA (Ubiquitous Web Applications)
 - * OOH (Object-Oriented Hypermedia Method)

Metodologías para desarrollo Web, II

- Ciclo de vida clásico. Pressman, “adapta” esta metodología para el desarrollo en la Web, lo que el llama la “Ingeniería Web”:
 - Formulación.
 - Planificación.
 - Análisis (contenido, interacción, funcional, configuración).
 - Diseño (arquitectónico, navegación, interfaz, de base de datos).
 - Implementación.
 - Pruebas.
- El método UWE (UML-based Web Engineering), propuesto por Koch et al., 2001.
- UWE consta de seis modelos:
 - Modelo de casos de uso para capturar los requisitos del sistema.
 - Modelo conceptual para el contenido (modelo del dominio).
 - Modelo de usuario: modelo de navegación que incluye modelos estáticos y dinámicos.
 - Modelo de estructura de presentación, modelo de flujo de presentación.
 - Modelo abstracto de la interfaz de usuario y modelo de ciclo de vida del objeto.
 - Modelo de adaptación.

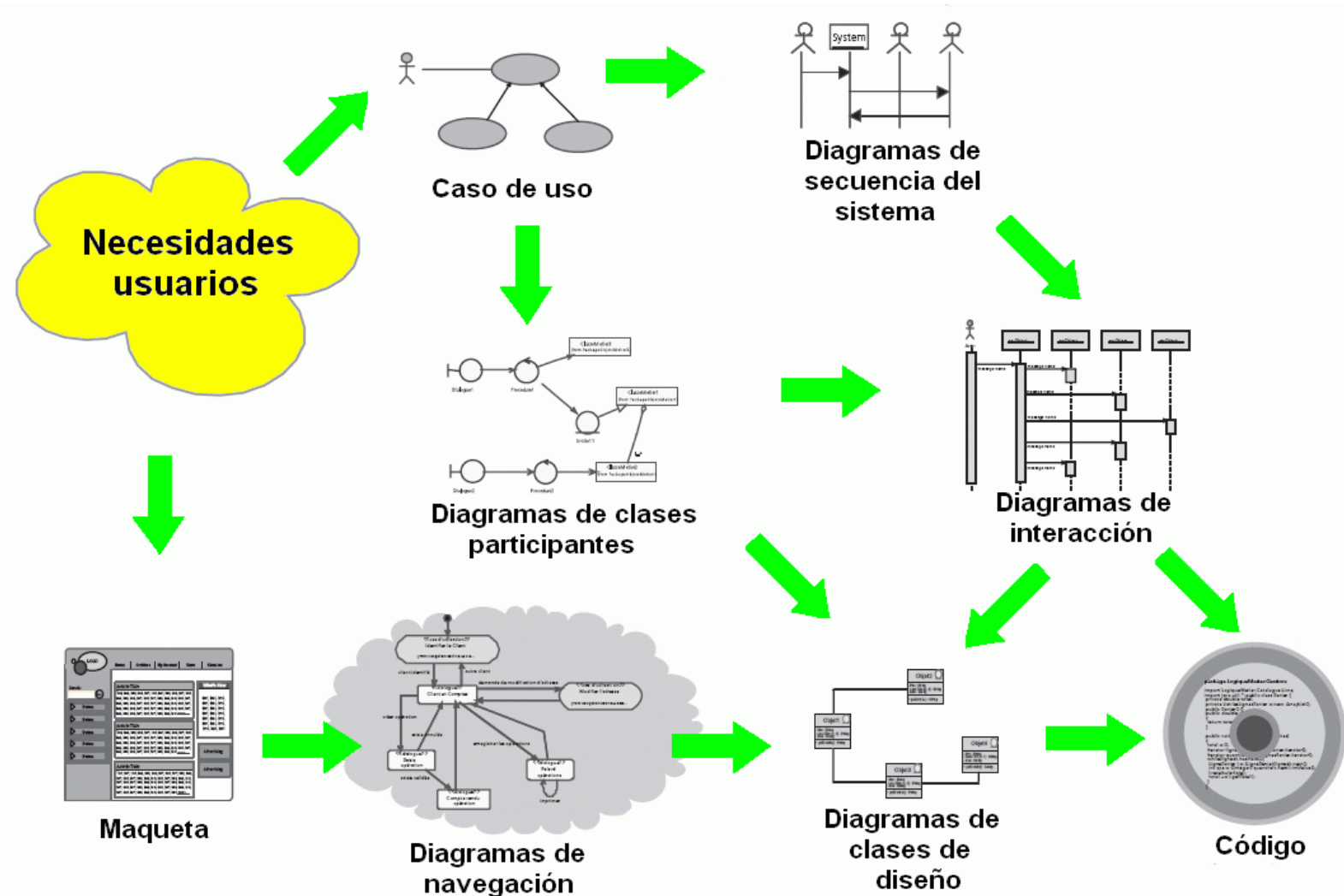
Metodologías para desarrollo Web, III

- Sánchez et al., 2007 propusieron una metodología de desarrollo Web ágil.
- El problema fundamental es el siguiente:
- Como pasar de los requisitos de los usuarios al código de la aplicación?
- En otras palabras: tengo una buena idea de lo que mi aplicación Web debe hacer, las funcionalidades esperadas por los futuros usuarios.
- ¿Cómo obtener lo más eficazmente posible un código computacional operativo, completo, probado, y que responda perfectamente a los requisitos? .



- Este es un esquema completo que muestra cómo, partiendo de los requisitos de los usuarios formalizados por los casos de uso y una maqueta de IHC, y con la contribución de los diagramas de clases participantes; se pueden conseguir diagramas de diseño a partir de los cuales se derivará el código de forma bastante directa.

Metodologías para desarrollo Web, IV



Aspectos de seguridad, I

- Las aplicaciones Web están más expuestas a ataques.
- Se pueden tener ataques en tres niveles:
 - A la computadora del usuario.
 - Al servidor.
 - A la información en tránsito.
- La seguridad en Web tiene 3 etapas primarias:
 - Seguridad de la computadora del usuario.
 - Seguridad del servidor Web y de los datos almacenados ahí.
 - Seguridad de la información que viaja entre el servidor Web y el usuario.
- Los usuarios deben contar con navegadores y plataformas seguras, libres de virus y vulnerabilidades. También debe garantizarse la privacidad de los datos del usuario.
- Se debe garantizar la operación continua del servidor, que los datos no sean modificados sin autorización (integridad) y que la información sólo sea distribuida a las personas autorizadas (control de acceso).

Aspectos de seguridad, II

- Garantizar que la información en tránsito no sea leída (confidencialidad), modificada o destruida por terceros. También es importante asegurar que el enlace entre cliente y servidor no pueda interrumpirse fácilmente (disponibilidad).

Recomendaciones: **Asegurar el servidor**

- Se deben considerar los siguientes puntos:
 - Asegurar el servidor en una forma fundamental: el sistema operativo, ya sea por medio de actualizaciones (parches) y habilitando los mecanismos propios de la plataforma.
 - Garantizar la seguridad del servidor Web propiamente (IIS, Apache, etc.)
 - Auditar las aplicaciones que interactúan en las dos capas anteriores (módulos, bibliotecas).

Recomendaciones: **Asegurar la información en tránsito**

- Esto se puede lograr por diversos medios:

Aspectos de seguridad, III

- ❑ Asegurando la red físicamente (switches en lugar de hubs).
- ❑ Esconder la información (esteganografía).
- ❑ Cifrar la información (criptografía) por medio de diversos algoritmos (SSL, VPNs).

Recomendaciones: **Asegurar el equipo del usuario**

- Vulnerar el equipo del usuario quizás no tenga el impacto de vulnerar el servidor, sin embargo es un problema más difícil de erradicar (1 servidor, 5000 clientes):
 - ❑ Aplicar actualizaciones (parches) al sistema operativo.
 - ❑ Uso de antivirus, firewalls personales.
 - ❑ Educación de los usuarios.