# *GUÍA PARA EL TRABAJO AUTÓNOMO*

*TEORÍA*

# TEMA 1. INTRODUCCIÓN A LOS SISTEMAS DE INFORMACIÓN BASADOS EN WEB

Objetivos:

- Conocer la historia y evolución de Internet y los Sistemas Web
- Identificar las particularidades del software desarrollado como Sistema Web
- Comprender la importancia de los Sistemas de Información Basados en Web

## UNA BREVE DEFINICIÓN DE INTERNET

> **Internet:** *Red informática mundial, descentralizada, formada por la conexión directa entre computadoras mediante un protocolo especial de comunicación. (Diccionario de la RAE)*

A la hora de afrontar la historia de Internet, quizá el mayor avance de la humanidad desde la invención de la imprenta, nos quedaríamos bastante cortos si nos limitáramos solamente al proyecto ARPANET o al famoso desarrollo de Tim Berners-Lee de la World Wide Web.

En 1995, el FNC norteamericano (Federal Networking Council), definió Internet como:

> *El sistema de información global que:*
>
> - *Está conectado lógicamente mediante un espacio de direcciones global y único que se basa en el protocolo IP o sus extensiones y derivados.*
> - *Es capaz de comunicarse usando el protocolo TCP/IP.*
> - *Proporciona, utiliza o permite acceder, pública o privadamente, a información y servicios de alto nivel que usan las comunicaciones e infraestructuras relacionadas previamente.*

En la definición anterior aparecen diversos conceptos relacionados con distintas disciplinas de la Informática y las Telecomunicaciones. En esta asignatura no vamos a centrarnos en los protocolos de red que sustentan Internet, pero sí tocaremos ligeramente los conceptos de espacios de direcciones (dominios, DNS) que son imprescindibles para conocer el funcionamiento de las aplicaciones web.

El desarrollo de Internet tiene dos pilares que se sustentan mutuamente:

- El hardware y los protocolos de comunicaciones
- El software y los estándares de estructuración de la información.

Aunque en la prensa, las cafeterías y en las conversaciones familiares se suele hablar de Internet y la Web como sinónimos, los conceptos son distintos.

**HISTORIA DE INTERNET**

Hay multitud de páginas dedicadas a glosar la historia de Internet, pero procede irse a la sociedad que vela porque Internet siga siendo abierta, transparente y definida para que todos podamos disfrutar de ella: la **Internet Society**.

En http://www.isoc-es.org/breve-historia-de-internet/ puedes leer una breve historia de este gran avance para la humanidad. Una versión más amena y no menos fiel a la realidad: http://www.walthowe.com/navnet/history.html

**Ejercicio 1.** *¿En qué momento la sociedad civil empezó a hacer suyo el proyecto inicialmente militar?*

**Ejercicio 2.** *¿Por qué he afirmado tan taxativamente que Internet y la Web no es lo mismo? Explique lo que es cada cosa para que lo entienda un profano en la materia.*

El núcleo principal de la asignatura se centrará en el análisis, diseño y desarrollo de servicios de alto nivel que permitan acceder a información de forma distribuida utilizando como soporte una infraestructura como los navegadores web. Pese a ello, vamos a dar una breve pincelada recordatoria de cómo se estructura Internet.

## EL MODELO DE RED DE INTERNET

Las comunicaciones a través de la red Internet se organizan en varias capas, según define el protocolo TCP/IP, cada una de ellas resolviendo un problema distinto:

- Capa de aplicación.
- Capa de transporte.
- Capa de internet.
- Capa de acceso a red.

Como ya conocerán ustedes por su asignatura Fundamentos de Redes, la bondad de esta estructura de capas es que en cada nivel o capa nos olvidamos del resto, pues se supone que hacen bien su trabajo, de forma que a nivel de aplicación es como si no existieran las otras tres capas subyacentes, y trabajamos en comunicaciones *proceso a proceso.*

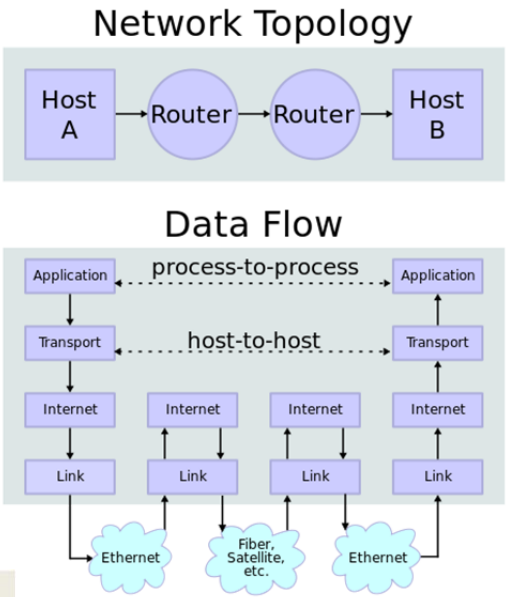


Ilustración 1: Flujo de datos según el protocolo TCP/IP [Fuente: Wikimedia]

**Ejercicio 3.** *Documente en sus apuntes las responsabilidades y protocolos usados en cada una de las cuatro capas indicadas.*

**Ejercicio 4.** *¿A qué capa pertenece el protocolo TCP? ¿Qué utilidad tienen los puertos?*

Es en la capa de aplicación donde nosotros vamos a trabajar en la asignatura, pues siendo una especialidad de Ingeniería del Software, no tiene sentido que nos centremos en aspectos de más bajo nivel. Sin embargo, sí es conveniente recordar cómo funcionan un protocolo de la capa de Internet, el protocolo IP, y dos protocolos de la capa de aplicación: el DNS y el HTTP.

## Capa de Internet: El protocolo IP

El protocolo IP identifica a cada elemento conectado a la red con una etiqueta numérica de forma unívoca, es decir, que dentro de una misma red no puede haber dos ordenadores con la misma dirección IP, pues no se sabría a qué destino entregar un mensaje. Según se utilice IPv4 o IPv6, las direcciones IP estarán formadas por 32 o 128 bits, y éstas pueden ser:

- Fijas. No cambia la dirección IP del dispositivo (salvo que se realice de forma manual) .
- Dinámicas. El dispositivo solicita una dirección IP cada vez que se conecta a la red a un servidor DHCP.

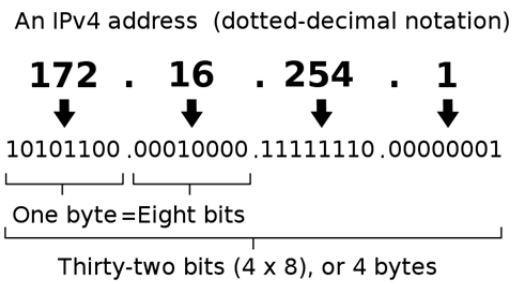


Ilustración 2: Dirección IPv4.

Los equipos que están permanentemente conectados a la red (servidores web, de correo, ftp, etc.) suelen tener IP fijas, mientras que nuestros routers ADSL o los equipos que se conectan a éstos suelen tener IP dinámicas. En el caso del router es nuestro proveedor de acceso a Internet quien dispone de un servidor DHCP, mientras que para la red local de nuestra casa, es el router quien realiza ese servicio.

Dentro del rango de direcciones IP posibles hay algunas reservadas para subredes privadas, esto es, para ordenadores que no están directamente conectados a Internet. Son aquellas en los intervalos:

- 10.0.0.0 a 10.255.255.255
- 172.16.0.0 a 172.31.255.255
- 192.168.0.0 a 192.168.255.255

Es la razón por la cual nuestros móviles, ordenadores, tabletas y demás dispositivos conectados a nuestro router ADSL tienen direcciones del tipo 192.168.x.x, mientras que nuestra IP "pública" es la que nos proporciona el proveedor de conexión (p.ej. 83.245.21.20).

### *IPv6*

La extensión a 128 bits fue necesaria porque las direcciones IP se agotaban en apenas unos años, y se ideó una extensión que permitiera la compatibilidad hacia atrás.

Una diferencia es que las direcciones ya no se expresan con números separados por puntos, sino con caracteres hexadecimales agrupados en bloques de 16bits separados por dos puntos.

Por lo demás, el cambio se realiza en los nuevos dispositivos de forma transparente para el usuario, y es de suponer que no habrá problemas en la adaptación de un protocolo a otro.

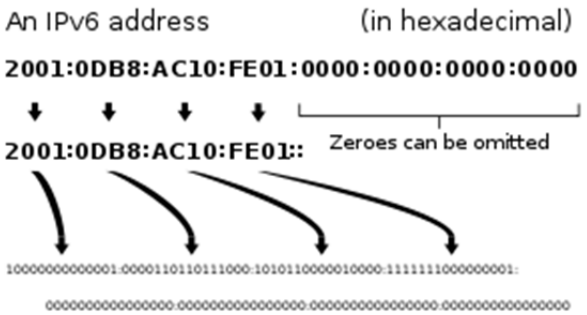


Ilustración 3: Dirección IPv6.

## Capa de Aplicación: El protocolo DNS

Si ya es difícil recordar un número de teléfono, o el número de cuenta del banco, los creadores de Internet se dieron cuenta que recordar direcciones IP iba a ser una tarea compleja. Por ello se ideó el protocolo *Domain Name System (DNS)*, que se encarga de traducir entre un conjunto de caracteres (nombre de dominio) y una dirección IP.

Un nombre de dominio se compone de dos o más partes (etiquetas) separadas por puntos.

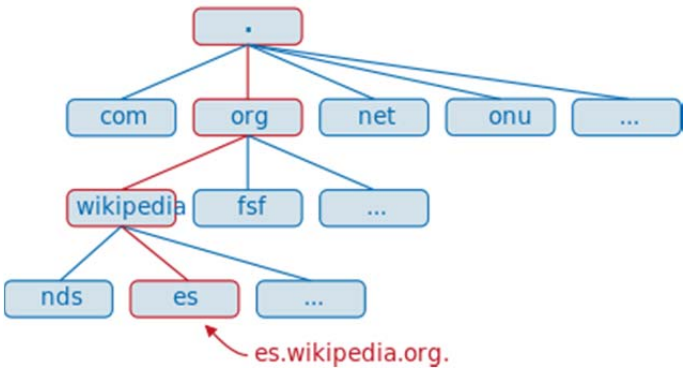


Ilustración 4: Jerarquía de nombres de dominio.

- La etiqueta más a la derecha se llama dominio de nivel superior o de primer nivel (.es, .com, .net, etc…).
- Cada etiqueta a la izquierda del dominio de nivel superior especifica una división o subdominio con respecto a lo que tiene a la derecha: esto es, **uma.es** o **ugr.es** son dos subdominios del dominio de primer nivel **.es**, de la misma forma que en ***agenciatributaria****.gob.es o* ***educacion****.gob.es,* los términos **agenciatributaria** o **educacion** son subdominios de **gob.es.**
- La etiqueta más a la izquierda suele corresponder con el nombre de la máquina, de forma que **etsiit**.ugr.es es la máquina **etsiit** del subdominio ugr.es, y **www**.ugr.es corresponde a la máquina **www** del mismo subdominio. A veces, especialmente en servidores compartidos entre varios alojamientos, diversos nombres de host e incluso diversos dominios apuntan a la misma IP, y es el servidor quien se encarga de acceder a una u otra parte del sistema de archivos.

Para configurar adecuadamente un servidor DNS, para cada subdominio hay que configurar una serie de registros, que permiten traducir de hosts a IPs concretas, incluso tener varios nombres para una misma máquina física. Estos registros son varios, siendo los que se modifican más habitualmente: A, CNAME, MX, TXT.

**Ejercicio 5.** *Investigue y documente cómo hacer que un ordenador de casa, conectado con IP dinámica, pueda hacer de servidor con un nombre de dominio fijo, de forma que siempre que tecleemos www.elordenadordemicasa.com estemos accediendo al servicio web que tengamos funcionando. Si dispone de dominio propio, utilice su ordenador de casa para ello.*

## Capa de aplicación: protocolo HTTP

Como bien sabe, la misión de la capa de aplicación es la de proporcionar los servicios para los que se establece la comunicación. Entre los servicios disponibles podemos enumerar o destacar el correo electrónico, el FTP y la mensajería instantánea.

La Web se basa en el protocolo HTTP (HyperText Transfer Protocol), que es un protocolo de nivel de aplicación que define un formato estándar para el intercambio de recursos en la web. Este protocolo se desarrolló a lo largo de la década de los 90 del siglo pasado con el objetivo de crear un sistema hipermedia distribuido que permitiera la comunicación cliente/servidor a través de una red TCP/IP.

Mediante HTTP un usuario que dispone de una aplicación cliente (p.ej. un navegador web) puede solicitar recursos de un servidor remoto (el *servidor web*). Los recursos habituales en este entorno son las páginas HTML, es decir, documentos multimedia con enlaces hipertextuales a otros documentos HTML. En realidad, de forma más general, la petición puede solicitar cualquier archivo de los alojados en el servidor, o incluso invocar la ejecución de un programa en el servidor.

HTTP tiene dos características muy importantes que hacen que los sistemas de información basados en Web se tengan que diseñar de forma muy distinta a las aplicaciones de escritorio:

- HTTP no está orientado a conexión, es decir, que cada petición es tratada de forma individual y sin un contexto. No existe la noción de sesión de usuario.
- HTTP es un protocolo de tipo “pull”, es decir, la interacción sólo se realiza en un sentido: cuando el cliente llama al servidor.

Afortunadamente, a lo largo del tiempo se han suplido las carencias de HTTP con otras tecnologías que permiten la gestión de sesiones, la interacción dinámica cliente/servidor, etc.

¿Cómo funciona el HTTP? En la ilustración 5 podemos verlo de forma esquemática. Cuando se solicita un recurso, se genera una petición HTTP. Esta petición se puede realizar siguiendo alguno de los métodos definidos por el estándar: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT y PATCH.

¿Cómo se localizan los recursos? Por su URL (*Uniform Resource Locator*), que no es más que una cadena de texto que contiene el protocolo a usar ( p.ej. HTTP o HTTPS para páginas web, FTP o SFTP para transferencia de archivos, etc.), el nombre o la IP de la máquina donde está el recurso al que queremos acceder, un número de puerto (opcional) y la ruta donde se encuentra el elemento que queremos recuperar. Adicionalmente se pueden pasar parámetros a continuación del nombre de archivo.

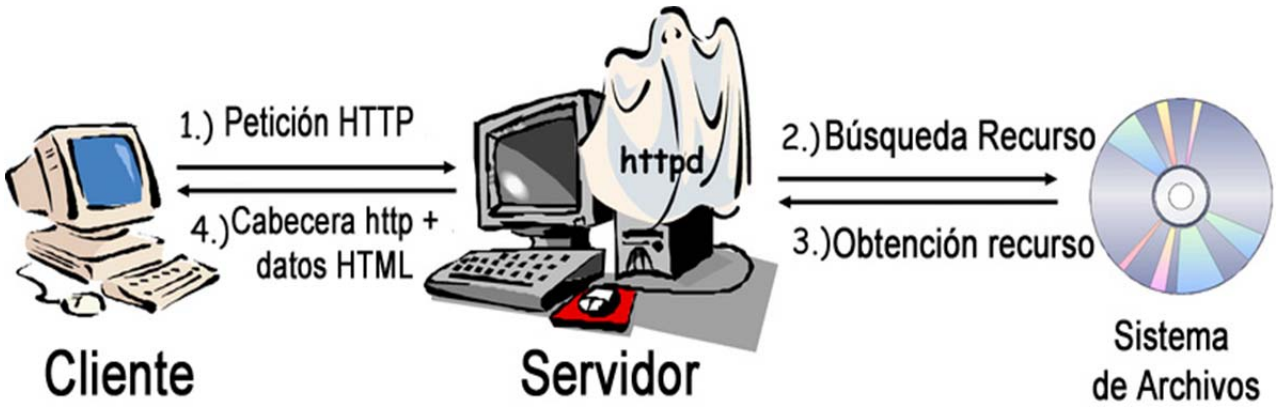


Ilustración 5: Esquema de obtención de una página web por protocolo http.

Por ejemplo, si escribimos en nuestro navegador

http://lsi.ugr.es/lsi/system/files/guias_docentes/SIBW_1516_0.pdf

estamos queriendo decir:

> *"quiero obtener del servidor lsi.ugr.es mediante protocolo HTTP el archivo SIBW_1516_0.pdf que se encuentra en la ruta /lsi/system/files/guias_docentes/ "*

Ojo, la ruta puede no referirse al sistema de ficheros real de la máquina, sino que se puede referir sólo a una parte concreta del árbol de directorios. Quien sabe eso es el demonio *httpd* y sus archivos de configuración.

Cuando en un navegador de internet se teclea

```
http://grados.ugr.es/informatica/index.php
```

el navegador interpreta la información según hemos explicado anteriormente y lanza el siguiente mensaje al ordenador grados.ugr.es:

```
GET /informatica/index.php HTTP 1.1
```

Tras recibir este mensaje e interpretarlo, el servidor envía un mensaje de respuesta. Dicho mensaje consta de tres partes:

- Una línea de status de la petición. Es un código numérico que indica el resultado: 200 "OK", 404 significa "Not found", etc.
- Un conjunto de cabeceras opcionales, con información como la fecha de generación del documento, o incluso la orden de crear una *cookie* con unos datos concretos.
- El cuerpo del mensaje, que normalmente es un conjunto de caracteres de texto con las etiquetas HTML.

**Ejercicio 6.** *Investigue y documente cómo obtener via telnet una página web.*

Los métodos se pueden clasificar como *seguros* o *inseguros* en el sentido de si modifican o no los recursos del servidor.

- *Métodos seguros* son HEAD, GET, OPTIONS y TRACE, ya que tan sólo se dedican a recuperar información del servidor.
- *Métodos inseguros* son POST, PUT, DELETE o PATCH, ya que permiten que el servidor reciba información para su almacenamiento, crear recursos o eliminarlos.

Los más importantes y frecuentes en el desarrollo de Sistemas Web son los métodos GET y POST. Aunque algunas veces se pueden usar indistintamente, lo normal es usar GET para recibir datos del servidor y POST para enviar.

- GET. Envía una solicitud en texto plano al servidor y permite al usuario, mediante una cadena de consulta (*query string),* enviar datos sencillos que sirvan para afinar la respuesta del servidor, a modo de parámetros, o para ser introducidos en la base de datos. Por ejemplo:
  http://www.sibw.es/ruta/script.php?nombre=valor1&apellidos=valor2
  invoca a script.php con dos parámetros, nombre y apellidos.

Otras características de las solicitudes GET:

- Se almacenan en caché y en el historial del navegador.
- Se pueden marcar como favoritos.
- No se deben usar con datos sensibles.
- Tienen restricción en cuanto a la longitud.
- Sólo se deben usar para obtener datos, no para enviarlos al servidor.

- POST. Envía una solicitud al servidor con datos adjuntos. La misma petición anterior se podría hacer con la URL http://www.sibw.es/ruta/script.php y enviar los parámetros en el cuerpo de la solicitud HTTP.

  Otras características del POST:
  - No se almacenan en la caché ni en el historial.
  - No se pueden guardar en favoritos.
  - No hay restricciones en cuanto al tamaño del adjunto.

A simple vista, POST parece un método más seguro, pues no se transmite la información en la propia URL. Además, resulta más limpio a la hora de construir las URL.

Ahora mismo esto les puede resultar extraño, pero es fundamental conocer la diferencia a la hora de implementar formularios y recuperar la información en el servidor.


Para que una máquina acepte peticiones HTTP no sólo es necesario que tenga una IP conocida y que esté conectada a la red, sino que debe tener un proceso, denominado demonio httpd que esté atento a las peticiones que le lleguen y sepa responderlas adecuadamente. En esta asignatura vamos a utilizar **Apache**, que es el sistema de servidor web más extendido en el mundo y además, es libre y multiplataforma.

Pero no sólo existe Apache. Se puede usar Microsoft Internet Information Services (IIS), Nginx, AOLserver… o cualquier programa que esté atento al puerto 80 TCP. Últimamente está de moda **node.js**, que no es más que una librería Javascript que trae implementado ese gestor de puertos http.

## LA WORLD WIDE WEB, WWW

La WWW tiene su origen en los trabajos de Tim Berners-Lee y Robert Cailliau en el CERN suizo, el laboratorio europeo de investigación nuclear. Inicialmente se creó como una forma fácil de distribuir y presentar información relevante para los físicos que allí trabajaban. Fue el precursor de diversos estándares, como URI, HTTP o HTML.


Ilustración 6: El primer servidor Web (CERN)

En los primeros años, la WWW era un elemento pasivo, es decir, una persona o entidad se encargaba de colocar la información que consideraba oportuno que los usuarios vieran, es decir, era una herramienta para publicar datos. De hecho, el primer navegador, mostrado en la ilustración 7, no tenía barra de direcciones.

El gran logro de Sir Berners-Lee fue recompilar conceptos ya preexistentes (como el hipertexto), y procotolos ya desarrollados (TCP/IP, DNS, etc.) y crear una herramienta de difusión de la información útil y escalable.

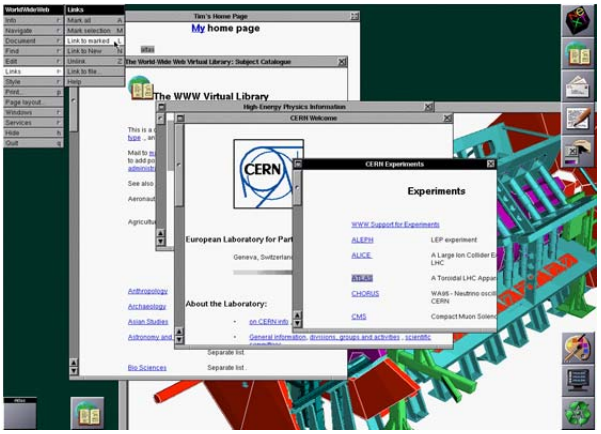


Ilustración 7: Primer navegador denominado "WorldWideWeb"

El W3C (World Wide Web Consortium) es una organización sin ánimo de lucro, liderada por Tim Berners-Lee, que se encarga de coordinar y desarrollar los protocolos sobre los que se basa la Web. Uno de sus principales objetivos es hacer accesible la WWW a todos y para todo.

En la web del W3C se pueden encontrar numerosos recursos didácticos para aprender todo lo relacionado con la tecnología Web:

http://www.w3.org/standards/webdesign/

http://www.webplatform.org/

http://www.w3devcampus.com/

**Ejercicio 7.** *Lea la autoentrevista que Tim Berners-Lee ofrece en la web del W3C y destaque lo que más le ha llamado la atención:*

*http://www.w3.org/People/Berners-Lee/FAQ.html*

**Ejercicio 8.** ***(Optativo)***

*Lea el artículo original en el que se "crea" la WWW, disponible en* http://www.w3.org/History/1989/proposal.html y haga un resumen

## CARACTERÍSTICAS Y REQUISITOS DE UNA APLICACIÓN WEB

Como hemos dicho anteriormente, las aplicaciones Web tienen una serie de características que no se encuentran en las aplicaciones software tradicionales:

- ***Mayor accesibilidad de información y servicios.*** Comparada con una aplicación de escritorio, la Web permite acceso a más usuarios de forma simultánea, además de que éstos posiblemente requieran que los datos y servicios se muestren de forma personalizada.
- ***Interfaz de hipertexto.*** La información debe mostrarse en un documento hipertexto, es decir, que haya interconexión entre distintas vistas de la información y las páginas, por lo que es necesario realizar un mayor nivel de abstracción y una adecuada organización de los datos y el flujo de trabajo.
- ***Diversidad de tecnologías para la gestión de la información.*** En la Web hay datos en muy diversos formatos: bases de datos tradicionales, XML, RDF, etc. Los ingenieros de desarrollo de sistemas web deben tener muy en cuenta cómo se accede a datos externos y su interacción con los datos propios.
- ***Diversidad de herramientas de visualización.*** La herramienta software desarrollada se va a mostrar en diferentes dispositivos (PC, Tableta), con distintas resoluciones de pantalla y también con diversos navegadores que posiblemente no interpreten exactamente igual nuestro interfaz.
- ***Complejidad de la arquitectura.*** El gran número de accesos y la escasa carga computacional de los clientes (los navegadores) requieren una arquitectura distribuida para el acceso a la información y los servicios.

Estas características conllevan a una serie de problemas o requisitos que hay que solventar a la hora de diseñar una aplicación web:

- **Concurrencia**. Es relevante este requisito pues puede darse el caso que un usuario modifique la información disponible para otros que estén accediendo a la vez al mismo recurso (p.ej. reserva de vuelos)
- **Carga impredecible**. En la mayoría de los casos no sabremos cuantas decenas, cientos o miles de personas estarán accediendo simultáneamente al servidor. Hay que preparar el software y el hardware para lo peor, dentro de unos límites razonables.
- **Rendimiento.** Si una web es muy lenta, todos sabemos lo que nos pasa: nos desesperamos, la abandonamos y no volvemos. Esta lentitud puede darse en el lado cliente o, principalmente, en el servidor. Las aplicaciones web han de desarrollarse para que la respuesta a las acciones del usuario sean casi inmediatas.
- **Disponibilidad.** Un sistema de información web debe estar disponible, idealmente, el 100% del tiempo. Nunca sabremos cuándo se conectarán los usuarios para hacer uso de él.
- **Orientación a los datos.** Las aplicaciones web suelen realizar tareas de consulta y modificación de bases de datos, no tanto cálculos científicos o financieros. Además, suele ser un requisito el mantenimiento de un log de eventos, que también son datos.
- **Evolución continua.** Las plataformas web se quedan obsoletas, algunas incluso en cuestión de horas (pensemos en un diario digital). De hecho, puede que al inicio del proyecto haya unos contenidos y días antes de la presentación éstos hayan cambiado completamente.
- **Inmediatez**. Este requisito se traduce también por “lo quiero para ayer”. Cuando un cliente nos solicita un desarrollo, normalmente lo quiere para “ya” pues espera tener una ventaja competitiva con respecto a sus rivales en el mercado.

- **Seguridad**. Los datos ya no se quedan estancados en nuestra memoria principal y en el disco, sino que atraviesan numerosos nodos intermedios. Las aplicaciones web requieren un extra de seguridad para que no caigan los datos en manos de cibercriminales.
- **Estética**. Si un desarrollo web no es agradable estéticamente no tendrá éxito, por muy funcional que sea.

Lo que diferencia a un ingeniero informático de un diseñador gráfico es que éste último se preocupa únicamente del último aspecto, la estética, mientras que los ingenieros debemos estar preocupados del resto de requisitos que pueden hacer que una web muy elegante estéticamente no sirva porque está continuamente caída, no evoluciona o da información errónea.

En el diagrama de la ilustración 8 podemos ver los diversos elementos en los que un desarrollador web debe estar pendiente, y si uno falla, el sistema empezará a fallar inexorablemente. El usuario final sólo percibe el resultado, en este caso www.ugr.es, pero el ingeniero que la ha desarrollado ha tenido que pensar en todos y cada uno de los elementos detallados.

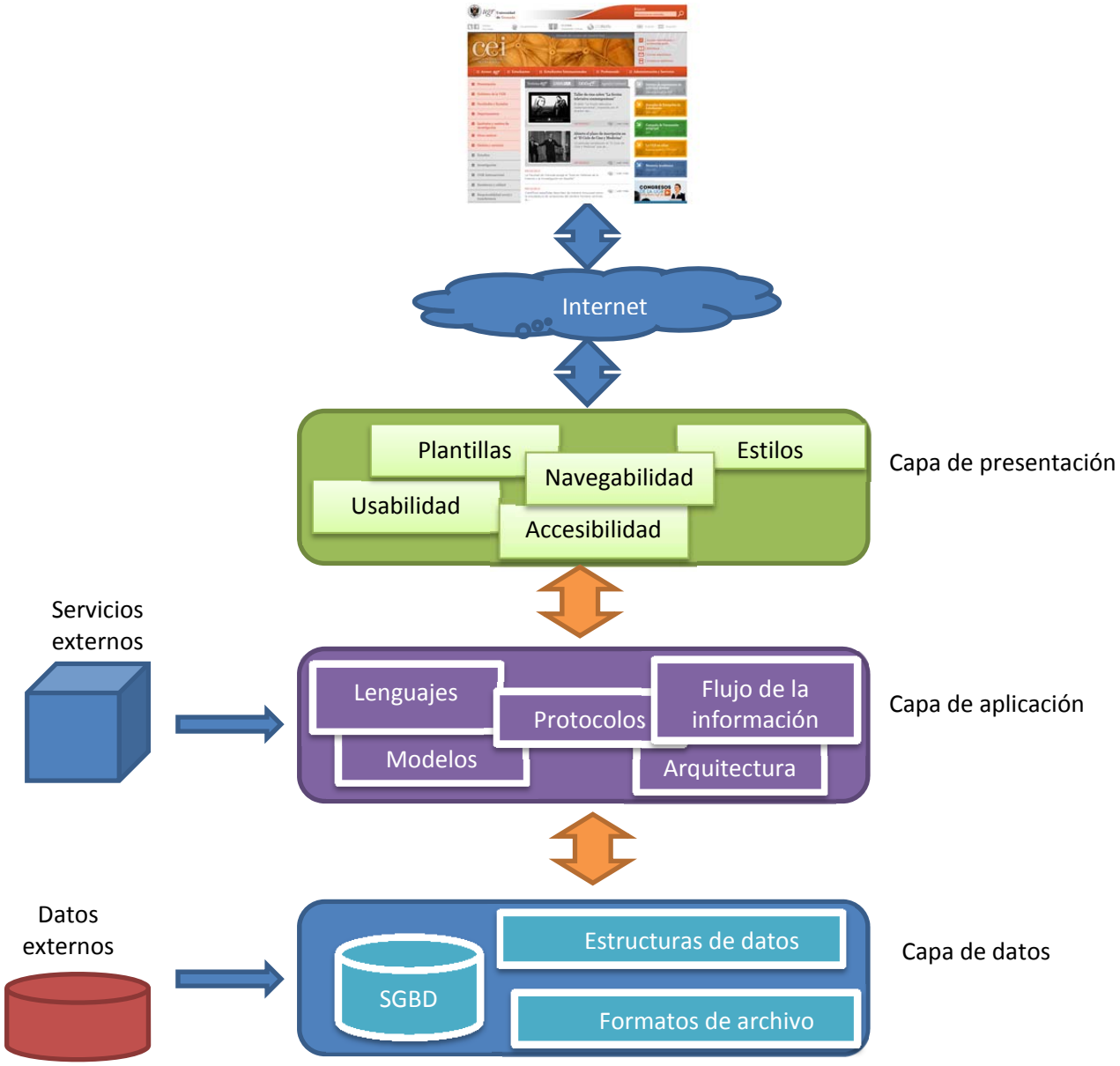


Ilustración 8: Los elementos de desarrollo de una aplicación web.

## UNA CLASIFICACIÓN DE LOS SISTEMAS WEB

Como bien supone usted, no todas las webs son iguales. No es lo mismo www.amazon.es que www.ideal.es, www.ugr.es o www.facebook.com. Sus objetivos, propósitos, funcionalidades y características son muy distintos y ésta es la primera decisión que hay que tomar antes de ponerse a desarrollar una aplicación basada en web.

Haciendo un repaso por la web, uno puede encontrar webs que se puedan clasificar en una o varias de las siguientes categorías:

- W*ebs informativas o de presencia.* Lo único que contienen es un mínimo contenido estático que permite mostrar productos y servicios. Hoy en día, cualquier pequeña empresa intenta tener presencia en Internet, y la primera aproximación es esa.
- *Webs de descargas.* Su objetivo principal es proporcionar documentos en formato distinto a HTML para su descarga por los visitantes. Podemos pensar en webs de grupos musicales que ofrecen sus composiciones gratuitamente, de fundaciones que ponen a disposición del visitante los libros que generan, etc.
- *Webs interactivas.* El usuario interacciona con la página, genera contenido y se pone en contacto con otros usuarios. Ejemplos claros son los foros temáticos.
- *Webs de contacto con el usuario.* Este tipo de webs son aquellas que permiten al usuario generar mensajes dirigidos al propietario de la aplicación. Pensemos en las aplicaciones de control de *bugs* o las de gestión del servicio de atención al cliente de un proveedor de servicios en internet.
- *Webs orientadas a servicios*. Una aplicación web orientada a servicios implementa el final de un servicio web, y genera la salida de éste. Pensemos en una aplicación web de reserva de vuelos; detrás del interfaz web HTML hay un complejo nudo de aplicaciones ejecutándose en el servidor que darán como resultado la modificación de uno o varios sistemas informáticos donde queda registrado el billete que hemos comprado.
- *Portales*, que no son más que sitios webs colectores de información y enlaces a otros sitios interesantes y relacionados con la temática.
- *Web alternativas a herramientas de escritorio.* El futuro es el *cloud computing*, el almacenamiento de los datos en eso que se viene a llamar *la nube* y que permitirá realizar todo lo que actualmente hacemos en escritorio a través de un navegador web.

Hay que saber distinguir entre un ***Sitio Web*** y un ***Sistema de Información Basado en Web***: el primero no permite al usuario modificar el estado del sistema, mientras que un SIBW sí.

Para dejarlo más claro: un buscador no es un SIBW, una aplicación web de gestión de la facturación de una empresa sí.

**Ejercicio 9.** *Indique al menos tres ejemplos de sitios web de las categorías descritas anteriormente.*

Esta categorización descrita es importante pues en función de los requisitos que nos marque el cliente, y por tanto el tipo de trabajo que nos solicite, habrá unas tecnologías más adecuadas que otras. En los próximos temas abordaremos el desarrollo de SIBW desde un punto de vista metodológico, tecnológico y normativo, pues es como un ingeniero debe desarrollar sus productos.