

<p>004.738.5:004.4 BLA pri</p>	<p>BLÁZQUEZ OCHANDO, Manuel Principios de Arquitectura de la Información / Manuel Blázquez Ochando .– Madrid: mblazquez.es, 2015. 109p. ; 21cm.– (Libros y manuales de la Documentación; 5) 978-84-608-3162-4</p> <p>1. Biblioteconomía y Documentación I. Título II. Series</p>
---	--



UNIVERSIDAD COMPLUTENSE DE MADRID
Facultad de Ciencias de la Documentación

1ªed. noviembre 2015, Madrid

© Copyright 2015. Manuel Blázquez Ochando

Publicado por mblazquez.es

978-84-608-3162-4

Índice

1. Introducción	3
2. ¿Qué es Arquitectura de la Información?.....	4
3. Necesidades de información y comportamiento del usuario.....	7
4. Sistemas de organización de contenidos	12
5. Sistemas de navegación.....	18
6. Sistemas de búsqueda.....	21
7. Representación de resultados e interfaz de búsqueda	28
8. Usabilidad web.....	33
9. Diseño de estructuras web con capas y CSS: layouts	38
10. Estilo de enlaces, menús y vinculación de archivos CSS.....	46
11. Estructura de carpetas y archivos	54
12. Maquetación de textos con CSS.....	57
13. Diseño de estructuras web: iframe vs include	62
14. Adaptación dinámica de las páginas de contenidos	68
15. Arquitectura de formularios y su adaptación dinámica.....	71
16. Navegación dinámica vs navegación estática	77
17. Gestores de enlazamiento y redirección para la navegación	81
18. Ejercicios prácticos	85
Práctica1. Deconstrucción de un sitio web.....	85
Práctica2. Planificación y desarrollo del sitio web.....	88
Práctica3. Editando las páginas de contenidos.....	92
Práctica4. Construyendo la estructura del sitio web.....	94
Práctica5. Adaptación dinámica de una página de contenidos.....	96
Práctica6. Controlando la información de los formularios.....	97
Práctica7. Creando la trama de navegación.....	98
Práctica8. Gestión, redirección y construcción de enlaces.....	100
Práctica9. Implementación de un buscador básico.....	102
Práctica10. Test de usabilidad.....	105
19. Bibliografía y referencias	106
20. Índice de tablas.....	107
21. Índice de figuras	109

1. Introducción

Podemos definir la arquitectura de la información como la técnica y metodología que permiten la construcción y edificación de productos informativos y documentales. Esto significa que el orden, la presentación y la pertinencia de los datos y contenidos publicados en muy diversos medios y soportes son los factores que intervienen directamente en esta obra. El objetivo es facilitar la lectura y comprensión de tales contenidos para cada tipo de usuario, anticipándose al planteamiento de sus necesidades informativas.

Por otro lado es frecuente observar el término arquitectura de la información relacionado o referido a la construcción de sitios web y documentos electrónicos véase (MARTÍN FERNÁNDEZ, F.J.; HASSAN MONTERO, Y. 2003). Esta acepción, mucho más moderna aún continua en constante evolución y por definir en cuanto su alcance, ya que entronca directamente con el segundo apartado de importancia en la asignatura, "*Desarrollo Web Avanzado*". La gran cantidad de cambios y progresos a los que se viene sometiendo la web dinámica, hace que los tipos documentales con los que se clasifican los contenidos de ayer queden obsoletos mañana. Dada la capacidad de reinención e interpretación de la realidad se descubren nuevos métodos de comunicación, representación y visualización de la información y en definitiva documentos cada vez más elaborados, con mayor significado contextual y más complejos en cuanto a su programación y edificación.

De esta forma la arquitectura de la información trata de aportar una manera de actuar, trabajar y programar la información, en los distintos tipos documentales y en especial a los referidos y consultados de desde la red. La dimensión multiplicadora de los documentos electrónicos y su universalidad para su redifusión, hace que el acceso, la facilidad de uso, la legibilidad, la claridad del diseño y la esquematización, la correcta narración, propiedad del lenguaje y semántica de los términos utilizados en los textos (HASSAN MONTERO, Y. 2002), sean propiedades claves de nuestra materia.

2. ¿Qué es Arquitectura de la Información?

- Es diseñar entornos de información para ser compartidos con los usuarios.
- Es el empleo y configuración de los sistemas de etiquetado, navegación, búsqueda y menús que permiten la organización de un sitio web o intranet.
- Es el método científico-práctico relativo a los productos de información, su constitución, configuración desde el enfoque de la usabilidad, accesibilidad y la recuperación de información.

Cuál es la razón de ser de la arquitectura de la información

- El coste de encontrar la información
- El coste de no encontrar la información
- El valor de educar al usuario
- El coste de construir un sitio web
- El coste de mantener el sitio web
- El coste de enseñar a encontrar la información en el sitio web

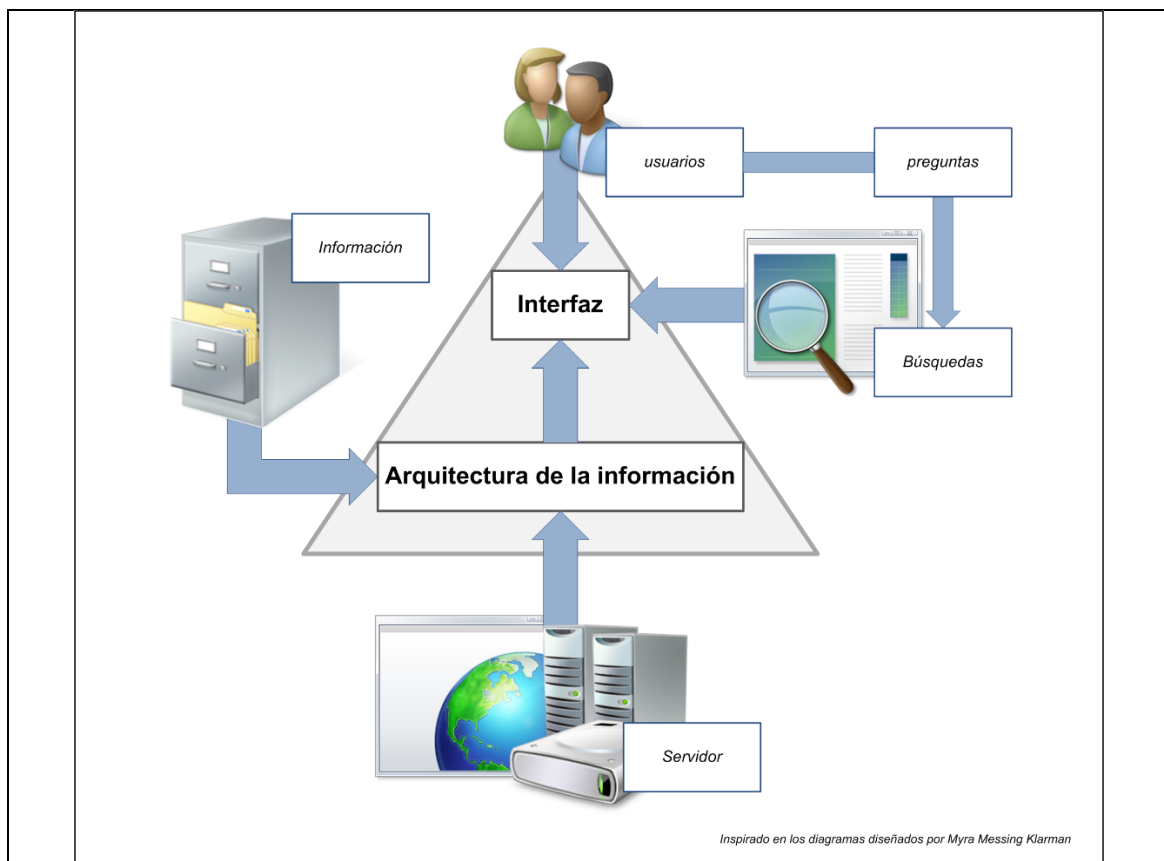


Figura 1. La Arquitectura de la Información en el centro de los servicios documentales

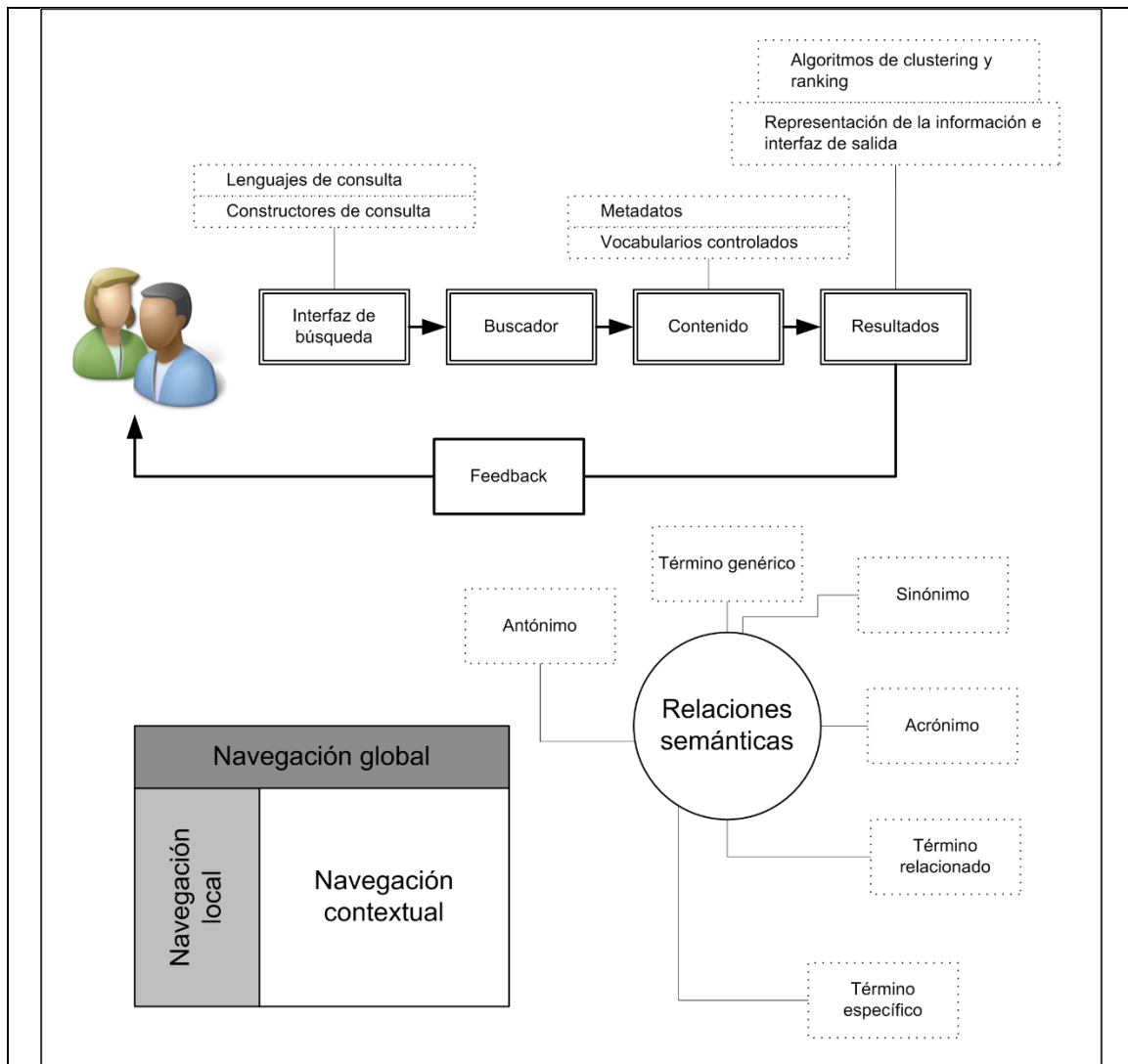


Figura 2. Diferentes entornos de la Arquitectura de la Información

Trabajos que implica la arquitectura de la información

- Bosquejos de sitios web
- Diagramas de contenidos y estructuras del sitio web
- Vocabularios controlados
- Esquema de metadatos

Capacidades del especialista en arquitectura de la información

- Diseñador de tesauros
- Editor de esquemas de búsqueda y consulta
- Especialista en metadatos
- Gestor de contenidos

- Gestor de las experiencias de los usuarios

Arquitectura de la información en el mundo real

- Definir el contexto
- Definir el contenido
 - Autor o responsable de los contenidos
 - Formato
 - Estructura
 - Metadatos
 - Volumen
 - Interacción o dinamismo
- Definir quiénes son los usuarios

3. Necesidades de información y comportamiento del usuario

El usuario de cualquier sistema de información plantea demandas y necesidades de información que se traducen en labores o actividades de búsqueda y recuperación de contenidos. Esta es la razón de ser de cualquier recurso o sitio web de información. Tales consultas se pueden clasificar en 4 tipos bien diferenciados:

- Búsqueda exhaustiva
- Exploración de contenidos
- Recuperación del elemento o contenido exacto
- Reformulación de la consulta o búsqueda reiterada

Las necesidades de información del usuario pueden variar a lo largo de la búsqueda y es por ello que se han elaborado diversas teorías para explicar este comportamiento. Por ejemplo la “*Berry-Picking Theory*” desarrollada por el Dr. Marcia J. Bates. Se determina que el usuario traduce sus necesidades de información en una cadena de texto, dicho de otra forma, formula una consulta, revisa la información devuelta por el sistema de forma iterativa, navegando entre los distintos resultados obtenidos, hallando datos que modifican o alteran la consulta original, perfeccionándola hasta lograr la información deseada. De esta forma el usuario no sólo logra encontrar los contenidos sino que aprende los mecanismos de refinamiento y recuperación del sistema.

Cómo aprender las necesidades y comportamientos del usuario

- **Análisis de la búsqueda.** Qué suele buscar en el sitio web, qué tipo de información. Esta información es almacenada en archivos de registro denominados Querylogs.
- **Investigación contextual.** Conocer en qué situación se efectúan las búsquedas y cómo el usuario interactúa con la información y herramientas que tiene a su disposición, tal como se muestra en el diagrama anterior.
- **Temática de la consulta.** Determinar la temática y acotar los contenidos para una mejor organización de la información. Utilizar lenguajes documentales especializados para cada temática ayuda el proceso de búsqueda y recuperación de información.

- **Grado de precisión y exhaustividad.** Cuánta información requiere el usuario y con qué precisión, ello conlleva un mayor control en su clasificación o una organización más o menos rígida, el tratamiento documental de los contenidos, su indización, etc.
- **Usos y aplicaciones de la información por parte del usuario.** En función de la calidad, extensión y precisión la información puede utilizarse para muy diversos fines. Conocerlos ayuda a preparar y presentar los contenidos de la forma más apropiada.

La experiencia del usuario y la recogida de bayas

The screenshot shows a Google search interface with the following annotations:

- Sistema de sugerencias de Google que se alimenta con las consultas de terceros usuarios:** Points to the search suggestions dropdown.
- Puede sugerir historia de la clasificación decimal:** Points to the search results for 'El sistema de clasificación decimal universal'.
- Puede reformularse la consulta con el término CDU:** Points to the search results for 'Clasificación Decimal Universal'.
- Los métodos de refinamiento, ayudan a reformular las consultas. La navegación a través de estas opciones facilita la resolución de las demandas informativas del usuario.** Points to the left sidebar navigation menu.

Below the screenshot is a flow diagram illustrating the user's information journey:

```

    graph LR
      Q1[Q1] -.-> Q2[Q2]
      Q1 -.-> Q3[Q3]
      Q2 -.-> Q3
      Q3 -.-> Q3.1[Q3.1]
      Q3 -.-> Q3.2[Q3.2]
      Q3.1 -.-> Q4[Q4]
      Q3.2 -.-> Q4
      Q4 -.-> Q5[Q5]
      Q5 --> Doc[Document]
  
```

Figura 3. Cómo la experiencia del usuario transforma los servicios de información

Estructurar la información

- **Organización.** Se deben establecer las categorías de contenidos, secciones principales del sitio web, de forma tal que el usuario comprenda en un primer vistazo, todo el abanico de contenidos que ofrece el sistema de información.
- **Sistemas de navegación.** Destinados a ayudar en la navegación de los contenidos en el sitio web. Esto es el uso de “*Breadcrumbs*” (Migas de pan), Directorios A-Z, menús contextuales, menús generales, etc.
- **Sistemas de búsqueda.** Utilizar un buscador en posición siempre visible con la configuración por defecto para su uso extensivo a todos los documentos del sitio, preparado para consultas rápidas.
- **Sistemas de etiquetado.** Titular todos los enlaces, clasificar los contenidos con sistemas de etiquetado mediante ontologías o descriptores libres o sociales facilita la navegación y recuperación final de los contenidos. En este sentido la aportación del usuario es clave para organizar la información.

Información permanente

- **Posición.** Dónde se encuentra el usuario en todo momento dentro del sitio web.
- **Desplazamiento.** Entre páginas mediante enlaces secuenciales (paginador, página siguiente, página anterior, páginas o contenidos relacionados)
- **Navegación jerárquica.** Conocer el nivel de profundidad en el que se encuentra en cada momento el usuario.
- **Nubes de etiquetas.** Categorías temáticas que clasifiquen todo el contenido del sitio web.
- **Opciones.** Contacto, ayuda, acerca de y derechos de explotación de los contenidos.

Siempre visible en todo momento

- Título y subtítulo en el margen superior del sitio web, concretamente en el encabezamiento.
- Logotipo o imagen identificativa siempre visible en el encabezamiento.
- Titulación de los distintos menús del sitio web.
- Titulación de las secciones ubicadas debajo del encabezamiento.

- Fragmentos de texto que describan los contenidos de cada sección o menú. ¿Qué contiene? ¿A dónde se puede llegar? ¿Para quién es este contenido?
- Las opciones de navegación siempre visibles para evitar el desplazamiento del scroll de arriba a abajo.

Sugerir información al usuario

- Basándose en las consultas realizadas por terceros usuarios es posible determinar qué contenidos fueron más visitados, qué contenidos son más recurrentes, qué grupos de contenidos responden a una temática ya clasificada en nuestro sistema. Por tanto la arquitectura de la información se basa en la experiencia del usuario para generar otras nuevas que mantienen las corrientes y modas en cuanto a búsquedas y necesidades de información se refiere.

Componentes en arquitectura de la información

- Organización del sistema de información
- Etiquetado de los contenidos
- Sistemas de navegación
 - Sistemas locales de navegación
 - Sitemap
 - Tablas de contenidos
 - Índices del sitio
 - Guías del sitio
 - Atajos
 - Sistemas de navegación contextual
 - Ayudas de navegación
- Sistemas de búsqueda
 - Interfaz de búsqueda
 - Lenguaje de consulta
 - Generadores de consulta
 - Algoritmos de recuperación
 - Áreas de consulta
 - Resultados de la búsqueda
 - Representación y visualización de los resultados de la búsqueda

- Contenidos y tareas
 - Intitulación de todos los enlaces, etiquetas y elementos que conforman el contenido del sitio web.
 - Hipertextualidad de los textos, relacionando los conceptos y términos que en él aparecen.
 - Uso extensivo de metadatos.
 - Párrafos, resúmenes y trozos de texto a modo de cata del resto de contenidos bajo una sección, artículo o área del sitio web.
 - Listas ordenadas de párrafos
 - Identificadores o elementos que demuestren en qué sistema de información se encuentra el usuario (logotipo, estilo, plantilla general, etc.)
- Componentes invisibles
 - Vocabularios, ontologías, tesauros
 - Algoritmos de recuperación
 - Resultados basados en la experiencia del usuario

4. Sistemas de organización de contenidos

Sistemas de organización exacta y objetiva. Permiten la recuperación exacta cuando el objeto o documento que se busca es conocido. Son de fácil elaboración, y consulta, pero no responden a una búsqueda exhaustiva o imprecisa. Son los sistemas de clasificación alfabética, cronológica y geográfica.

- **Alfabética.** Es el método de organización primario que proporciona un índice alfabético de un dominio o área de conocimiento. En este caso de un sitio web completo o de alguna de sus secciones.
- **Cronológica.** Cuando la información se sucede en el tiempo tiene su razón de ser una recuperación por cotas temporales. Suele ser de habitual aplicación en noticias de prensa, titulares y canales de sindicación o cuyo flujo de información se sucede de manera continua.
- **Geográfica.** La organización geográfica permite limitar la información según su origen o ámbito de aplicación, lo que permite una discriminación de los contenidos que ayuda a la recuperación y búsqueda del usuario. Por ejemplo, distinguir la geolocalización de fotografías, los medios de comunicación de cada país o los servicios prestados por una empresa para cada región.

Sistemas de organización subjetiva. Implica una subjetividad y ambigüedad o inexactitud, puesto que el profesional de la información debe clasificar contenidos en función de categorías temáticas que no siempre responden a la definición de los contenidos, o al menos no completamente. Su aplicación es de gran importancia para responder a la mayoría de búsquedas del usuario, en las que desconoce el objeto exacto de su consulta, obteniendo una aproximación a partir de estos sistemas de organización.

- **Temática.** Uno de los métodos más extendidos es la clasificación temática basada en tesauro o taxonomía, lenguajes en todo caso controlados y normalizados. Si bien todo sitio web debería contar con este método, debería ser combinado con una presentación continua de novedades en cada uno de sus

apartados, para convertirse en un esquema de contenidos actuales. Este mecanismo ayuda a educar al usuario y formar una idea real de la información que enmarca cada área temática o sección.

- **Tareas.** Definir los servicios o actividades que puede desempeñar el usuario constituye otro tipo de clasificación subjetiva que suele estar combinada con la organización temática o de audiencia.
- **Audiencia.** Si existen varios tipos de usuario, puede tener sentido crear varios esquemas de información que respondan a cada caso, por ejemplo usuarios domésticos, profesionales, investigadores, educadores, cuyas demandas informativas variarán ostensiblemente. Por otro lado la posibilidad de interactuar entre los distintos entornos de usuario permite definir un sistema de organización más abierto o cerrado.
- **Metafórica.** El empleo de metáforas visuales para distinguir los elementos y contenidos de un sitio web, puede resultar de gran ayuda para comprender el contexto y educar al usuario en cuanto a su objeto de consulta. Es el caso de los mapas de imágenes, mini-sites con contenidos multimedia de tipo flash, o las infografías utilizadas para explicar distintos hechos, objetos y circunstancias.

Organización jerárquica. Las jerarquías, ya fueren tesauros o taxonomías proporcionan un método simple de clasificación y organización de la información, permitiendo enfocar la consulta del usuario desde el plano general hasta el más específico. En el caso de las taxonomías, se recomienda que las categorías jerárquicas sean en mayor medida excluyentes del resto de conceptos para evitar en la medida de lo posible la ambigüedad. Ello conlleva ajustar la precisión del lenguaje, para lograr un balance clasificatorio inclusivo equilibrado de los documentos. Dicho de otro modo, es recomendable mantener una especificidad y una capacidad de discriminación alta para que una clasificación de tipo polijerárquica. Por otro lado el control entre la amplitud (el número de elementos que conforman un nivel jerárquico de la taxonomía) y la profundidad (el número de niveles jerárquicos de la taxonomía) puede incidir en una clasificación demasiado horizontal o vertical, lo que dificulta la navegación entre sus contenidos. En este sentido también se requiere un equilibrio y una proporcionalidad.

Hipertexto. Consiste en un método de enlazamiento no lineal y altamente estructurado de la información. Ello implica que el trozo de texto que se enlaza (mediante URL) a un tercer contenido, encaja con su significación, sentido semántico, definición o explicación y ejemplificación. También puede existir una organización jerárquica entre los distintos trozos de texto y más concretamente, entre las páginas que conectan entre sí. No obstante, no se produce con un orden previo o precoordinado como ocurre con los lenguajes controlados, lo que proporciona gran libertad y flexibilidad de interrelación. Sus desventajas derivan de la falta de normalización y la subjetividad a la que están sujetas las relaciones (quedando a la elección y entendimiento del profesional de la información), lo que facilita que un usuario que navegue a través de este sistema se pierda fácilmente. Dicho de otra forma, las relaciones que se establecen entre documentos, puede que no sean comprendidas por todos los usuarios de la misma forma con que son entendidas por su creador.

Sistemas de clasificación social. También conocido como etiquetado social, etiquetado libre, etnoclasificación o colaborativo, consiste en un mecanismo que permite la interacción de los usuarios para clasificar los contenidos según su experiencia y conocimientos con la documentación y medios que utilizan en el sistema. Este mecanismo proporciona una gran libertad y extensibilidad para organizar la información, justamente todo lo contrario a una taxonomía o un tesaurus, lo que implica que la inteligencia clasificatoria no emerge del arquitecto de información sino de los propios usuarios o personas reales. A pesar de ser un método enormemente popular, también causa problemas a la hora de recuperar la información clasificada, dada la ambigüedad de los términos empleados en las etiquetas y al carecer de un mínimo control para la organización de su contenido. Dicho de otro modo, se hace difícil establecer la clave correcta de acceso a la información y con ello se corre el riesgo de emplear términos inexactos e incluso equivocados.

Sistemas de etiquetado. Los sistemas de etiquetado, tan frecuentemente utilizados en los sitios web, pueden plantear problemas de significación, navegación y referencia, si no se atiende al lenguaje que emplean los usuarios para comunicarse y navegar habitualmente en la red. Lo que haría preguntarse si las etiquetas utilizadas para las distintas secciones del sitio web, responden realmente a los contenidos que espera

encontrar el usuario, si estas ofrecen algún tipo de explicación o si es posible acceder a esa información explicativa extra de algún modo sencillo. Algunos errores más comunes son los siguientes:

- Etiquetas ambiguas, sin significación o demasiado comunes. Por ejemplo el término principal o home. Se entiende que alude a una página principal, pero no queda claro, puede ser la portada del sitio web, la página de entrada o de inicio.
- Uso de dos o más etiquetas simultáneamente. En el caso de querer unificar contenidos bajo un mismo epígrafe, se puede producir una confusión al respecto de la información contenida. Por ejemplo, productos y servicios, recursos y bibliografía, etc.
- Adecuación de la etiqueta a la realidad contenida. Utilizar términos que ofrezcan una visión real de los contenidos. Por ejemplo no utilizar términos como súper-ofertas, pero sí ofertas. Esto es adecuar el lenguaje a la realidad y al vocabulario controlado.
- El uso del lenguaje corporativo. En muchos casos el empleo de un lenguaje empresarial puede desconcertar al usuario, por ejemplo sinergias, estrategias, etc. Pero también dentro de dicho vocabulario existen etiquetas aceptadas y convenidas comúnmente, por ejemplo sobre nosotros, about-us, derechos, etc.
- Las etiquetas no sean representativas y no permiten distinguir o diferenciar.
- Las etiquetas se convierten en jerga no centrada en el usuario
- Las etiquetas no ofrecen una visión correcta de lo que contienen

Tipología de etiquetas

- **Enlaces contextuales.** Son trozos de texto enlazados a terceros recursos y contenidos en la misma página o terceras páginas. Tales enlaces y sistema de etiquetado es denominado Hipertexto.
- **Titulares - headings.** Se limitan a describir el contenido que encabezan. Permiten establecer una jerarquía dentro del texto permitiendo distinguir los distintos epígrafes que atañen a un artículo, libro, o página web. Los métodos de numeración, viñetas, negrita y tamaño de la fuente ayudan en este sentido a crear un orden de presentación de la información. Dichos titulares actúan como trozos de texto representativos y pertinentes de los contenidos, distinguiéndose dicha diferenciación de manera visual. El objetivo de estos es establecer un orden de

lectura, que permita saber el comienzo del contenido, su resumen, sus resultados, sus objetivos, su motivación, su desarrollo, etc.

- **Opciones de navegación del sistema.** Las etiquetas que representan las opciones de navegación del sistema o que están integradas dentro de él, constituyen una constante, en todas las páginas del sitio web, lo que requiere una racionalidad en la organización de los contenidos y de los conceptos que representan. No existe una convención o normalización en este aspecto pero sí se reconocen una serie de términos de uso común y altamente difundido en cualquier sitio web. En estos casos, siempre se recomienda emplear la meta-descripción de la etiqueta y de los enlaces que portan de tal forma que sean suficientemente significativos de las opciones que ofrece el sitio web.
 - página principal, inicio = main page, home
 - buscar, explorar, navegar = search, find, browse
 - contactar = contact us
 - ayuda, FAQ = help, FAQ
 - noticias, eventos = news, announcements
 - acerca de, sobre nosotros, quienes somos = about, about-us, who-we-are
- **Términos de índice.** Son palabras clave, encabezamientos de materia, descriptores que representan el contenido y que conforman puntos de acceso a la información propiamente dicha. Son utilizados para describir trozos de texto, encabezamientos, títulos, enlaces, opciones del menú de navegación, los contenidos propiamente dichos. Tales usos están destinados casi por completo a la recuperación de los contenidos publicados, siendo en muchos casos invisibles cuando aparecen en forma de código de metadato o meta-etiqueta.

Coherencia y consistencia del sistema de etiquetado. El etiquetado de contenidos es sencillo siempre y cuando el ámbito de aplicación, la temática, el contexto y los usuarios sean un conjunto concentrado bien delimitado desde un principio. Esto significa efectuar un etiquetado modular dependiendo de las secciones especificadas en el sitio web. Cada sección requiere de un vocabulario y un etiquetado especializado que permita navegar en cada caso con el lenguaje correcto. En todos ellos el factor clave para su funcionamiento es la coherencia que significa el uso de términos previsibles

para el usuario y fáciles de recordar en un primer golpe de vista general del sitio web. Ello depende de diversos factores:

- **Estilo.** Los signos de puntuación o caracteres que dificultan la lectura han de ser suprimidos o sustituidos por otros que presenten una puntuación sencilla y simple. También el empleo de guías de estilo facilita la definición del etiquetado.
- **Presentación.** El tamaño de la fuente, el tipo de letra, el color, su espaciado, su resaltado puede ayudar o no en aportar coherencia al conjunto. Por ello es recomendable utilizar estilos sencillos sin decoraciones, coherentes a lo largo de todos los contenidos y páginas del sitio web.
- **Sintaxis.** Utilizar un único método de construcción sintáctica para las frases, en todos los casos, ayuda a crear la coherencia necesaria en el sistema. Por ejemplo mantener una estructura sujeto-verbo-predicado, o verbo-predicado en el caso en el que se citen acciones o tareas concretas en el sitio web.
- **Granularidad.** Dentro del sistema de etiquetado es posible encontrarse términos que responden a distintos niveles jerárquicos dentro de una misma faceta, por ejemplo aviación civil, aeropuerto, aerotransporte. Distintos niveles dentro de una faceta en vez de ayudar dificulta la comprensión de la clasificación del contenido, requiriendo términos o etiquetas más específicas de un nivel en cuestión, reduciendo dicha granularidad.
- **Comprensión y coherencia: amplitud.** Si se describen distintos elementos que conforman una realidad dentro de una misma categoría y nivel granular, se ayuda a comprender el objeto y naturaleza del contenido. Por ejemplo calzado, bolsos, cinturones, bisutería es indicador suficiente de que se está en la categoría complementos.
- **Adecuación del lenguaje al tipo de usuario.** Se trata de adaptar el lenguaje al tipo de usuario. Por ejemplo, un usuario especializado puede considerar pertinente el concepto apnea, cuya traducción o nombre común sería trastorno del sueño.

5. Sistemas de navegación

Sistemas de navegación principales. Su objetivo es la navegación a través de todos los contenidos del sitio web, permiten establecer el contexto y la flexibilidad de navegación.

- **Global.** Debe indicar dónde se encuentra el usuario y a dónde puede ir.
- **Local.** Su objetivo es mostrar qué opciones o temas colindantes se encuentran dentro de la sección que está siendo consultada.
- **Contextual.** Trata de explicar qué contenidos hay relacionados con la página que se está consultando.

Sistemas de navegación complementarios. Sirven a un propósito concreto y están integrados dentro de los sistemas de navegación principal.

- **Mapas de sitio.** Corresponden a la organización de secciones o categorías del sitio bajo las que se adscriben los distintos enlaces a los contenidos o subcategorías.
- **Índices.** Consiste en una organización alfabética de todos los enlaces a las secciones, categorías y subcategorías del sitio web.
- **Guías.** Método de navegación que establece pasos o puntos de verificación entre los distintos contenidos del sitio web.

Errores muy graves en el diseño de los sistemas de navegación

- Pistas para que el usuario averigüe los enlaces visitados y no visitados. Implica controlar los colores de los enlaces en todos sus estados.
- Suprimir el botón de marcha atrás y opciones que no permitan al usuario volver sobre sus pasos.
- Paralizar la función de favoritos del navegador.

Construyendo el contexto del sitio web

- Indicar al usuario en dónde se encuentra en todo momento. Si es preciso indicar... usted está aquí. Tener o no ese indicador marca la diferencia entre una buena y una mala arquitectura de información.
- Marcar la posición o situación de cualquier contenido dentro del sitio web.

- Mantener los mismos esquemas de diseño en todo el sitio web. Logo, título o denominación del sitio o proyecto, etc.
- Si se utiliza un sistema de organización jerárquica éste deberá indicar la posición en la que se encuentra el usuario.
- Los niveles jerárquicos en los que se encuentran las categorías y subcategorías deben permitir una navegación entre iguales y niveles inferiores y superiores.
- Someter el sistema de navegación a una prueba de esfuerzo [1 <http://instone.org/navstress>] [2 <http://www.optimalworkshop.com/treejack.htm>]

Sistemas de navegación embebidos

- Sistema global de navegación.
 - Siempre presente en todas las páginas y contenidos del sitio web.
 - Forma de barra de navegación. Por lo tanto menú horizontal en el margen superior de la página web.
 - Permite el acceso a las funciones y secciones clave del sitio web.
 - Siempre proporcionan acceso a la página principal, buscador, título o nombre de la página en la que se encuentra el usuario,
 - Máximo 1 sistema de navegación global.
- Sistema local de navegación.
 - El sistema global de navegación es complementado por el local.
 - Puede haber más de 1 sistema local de navegación.
 - Ofrece información y subcategorías adscritas a la categoría o sección dependiente o principal que el usuario selecciona en cada caso.
 - Dependiendo del área que abarquen y su nivel de descentralización del sitio web, pueden llegar a ser considerados subsitios.
- Sistema de navegación contextual.
 - Potencia el aprendizaje asociativo de los términos enlazados a recursos, páginas relacionadas y sus contenidos. Los usuarios aprenden mientras navegan.
 - Permite generar un tejido colectivo de contenidos.
 - Constituyen el último sistema de navegación que se edita, ya que dependiendo de cómo se integran los contenidos en el sitio, se diseña un

hipertexto, un índice o una guía adaptada a cada caso, espacio, área e interfaz.

Los tres sistemas de navegación principales deben ser integrados y coexistir sólo cuando sea necesario para una correcta organización y navegación. La prioridad es facilitar el acceso a los contenidos y no abrumar al usuario. Esto implica mantener un equilibrio entre el número de opciones disponibles y el área o espacio que ocupan en la pantalla. Se recomienda el uso de menús textuales y no gráficos en la medida de lo posible, uso de atributos “*alt*” y “*title*”, evitar imágenes e iconos en los sistemas de navegación, ya que el texto es siempre mucho más claro. Evitar mecanismos complejos de presentación de las opciones del menú en la medida de lo posible, por ejemplo Flash, Java, JavaScript. Finalmente evitar el uso de marcos (excepto “*iframe*”) para cargar los contenidos enlazados desde el menú en beneficio del sistema de capas de tipo div y el empleo de codificación en lenguaje PHP, para estructurar las partes del sitio web.

6. Sistemas de búsqueda

Antes de instalar un buscador en el sitio web...

¿Tiene el sitio web suficiente contenido como para requerir un buscador?

- Determinar un número suficiente de páginas y contenidos para integrar un buscador. Por ejemplo entre 500 y 1000 páginas.
- Información muy especializada puede ser suficiente motivo para requerir un buscador, ya que esta puede ser demandada con frecuencia y en poco tiempo por el usuario.
- Si el sitio web tiende a parecerse a una biblioteca más que a un sitio web liviano, será muy necesario introducir el buscador que facilite la recuperación de la información.

¿Interesa más invertir recursos en un mejor buscador o en un mejor sistema de navegación?

- Un buscador no es la solución para los problemas de navegación de un sitio web. Antes de incorporar ningún buscador es necesario arreglar tales problemas.
- Cuando el sitio web consta de un lenguaje controlado para clasificar la información y documentación, la implantación del buscador es más sencilla y su funcionamiento más eficiente.

¿Tiene tiempo y conocimientos para optimizar el sitio de búsqueda?

- Un motor de búsqueda resulta sencillo de poner en marcha, pero su implementación y configuración no lo son tanto. En este sentido una buena aplicación e interfaz ayudan al usuario a controlarlo adecuadamente. Un aspecto y otro implican un trabajo de programación y edición integral del buscador para afinarlo a las necesidades del usuario.

¿Hay mejores alternativas?

- Cuando no se tienen los conocimientos técnicos necesarios para editar un buscador o configurarlo adecuadamente, o bien no se dispone de fondos para contratar a un especialista que lo haga, sólo quedan dos alternativas; 1) encontrar

un buscador más sencillo de instalar y configurar, 2) editar un índice de todos los contenidos del sitio web.

- Elegir bien el buscador es importante, no siempre el más potente es el que mejor se adapta a las necesidades del sitio web y de los usuarios o el que mejor relación sencillez/efectividad tiene. Por lo tanto el motor de búsqueda que se instale debería ser fácilmente instalable y configurable, tener una eficacia en la recuperación razonable y ser amigable en su interfaz y representación de resultados para el usuario.

¿Los usuarios prefieren navegar a buscar?

- Puede ocurrir que el tipo de usuario concurrente del sitio web prefiera navegar a utilizar el buscador, o bien el buscador es el último recurso. Esto ocurre cuando los sitios web no tienen un gran crecimiento. Pero cuando el sitio web crece desorbitadamente, puede llegar un momento en el que la navegación habitual pueda llegar a desaparecer hasta quedar sustituida por un prominente sistema de búsqueda, destacado de todo el sitio.

¿Se pretende crear un buscador para unir y enlazar sitios fragmentados?

- Cuando un sitio web está partido en diversos micrositos un buscador puede ser una buena solución para integrarlos y recuperar en cada uno de ellos indistintamente. Esto permite que el usuario conecte con temas o áreas que le son desconocidas, aportando un poco de coherencia a su periplo.
- No obstante aunque ello pudiera ser la solución, sólo lo es en parte para el usuario pero no para el administrador de todos los contenidos publicados en dichos sitios fragmentados.

Considerar el buscador como una herramienta de aprendizaje para el usuario

- Mediante sistemas de evaluación y recogida de datos, se puede estudiar el comportamiento del usuario durante su sesión de consultas con el motor de búsqueda. Recuérdese el método de Querylogs.

Si los usuarios esperan un buscador se deberá incorporar

- Aunque la tesis del administrador diga lo contrario, el usuario es quien manda a la hora de introducir los sistemas de búsqueda y recuperación de información. Si se espera dicho servicio, será necesario incluirlo.

Controlar la información y contenidos dinámicos

- Si los contenidos son muy dinámicos, puede ser imprescindible el uso de un buscador que permita al usuario encontrar el último minuto o la última hora de lo que se está publicando. Esto significa indexar los contenidos para una recuperación rápida y efectiva.

Elegir dónde buscará el motor

Determinar dónde debe buscar el motor de búsqueda puede implicar la indexación de todos los textos del sitio web o un determinado área, para ganar eficacia y especialización en la recuperación. Dependiendo de la heterogeneidad de los sitios web y de las bases de datos con las que están compuestos así dependerá la estrategia de selección de objetivos para el buscador. También existen otros aspectos que elegir, por ejemplo los formatos de datos que el sistema indexará (HTML, XML, PDF, DOC, PPT, etc.) o qué campos de la base de datos tomará en cuenta para almacenar y recuperar la información a texto completo.

Las zonas de búsqueda

- Son subconjuntos que han sido indexados de manera particular y por separado del resto del sitio web de tal forma que cuando el usuario navega y entra en el área o categoría específica, el buscador varía de contenidos generales a otros mucho más específicos, permitiéndole encontrar información de forma más precisa.
- Los factores que determinan las zonas de búsqueda son: el tipo de contenido, el público objetivo, las funciones del sitio, las categorías y subcategorías temáticas, el origen geográfico, la datación cronológica, el autor y el departamento, negociado, grupo de trabajo o unidad administrativa.
- No obstante el establecimiento de zonas de búsqueda puede tener contraindicaciones, por ejemplo puede entrañar dificultades de interacción

para el usuario ya que el contenido al estar parcelado requiere de un proceso de navegación y de consulta continuado.

Navegación y páginas de destino

- Los sitios web contienen dos grandes tipos de páginas. Por un lado aquellas destinadas a la navegación y redirección del usuario y aquellas páginas que portan el contenido que finalmente desea encontrar el usuario, también denominadas páginas de destino. Si la indexación del buscador se hace sobre todo el contenido del sitio web incluyendo las páginas de navegación, se puede producir un problema importante. La página de resultados se llenará de páginas de navegación, obstruyendo todo el proceso de búsqueda y será como una aguja en un pajar averiguar dónde se encuentra la página de destino con el contenido. Por lo tanto es muy recomendable regular el número de páginas de navegación que deberán ser indexadas a fin de que abunden más las páginas de destino.

La indexación de contenidos según el público objetivo

- Sin bien indexar contenidos en función de la tipología de públicos y usuarios del sitio web podría ser una ventaja, resulta de importancia también calibrar el impacto. Es posible que exista un índice de solapamiento entre los contenidos de los distintos usuarios lo que empeoraría la recuperación de información. Es aconsejable fijar en tal caso las páginas de destino unívocas para cada público para reducir dicho efecto.

Indexación por tema

- Resulta muy conveniente establecer la temática de los contenidos indexados, esto permite enfocar previamente la consulta del usuario y redirigirla a un área muy específica del corpus tratado por el sistema.

Indexación de los últimos contenidos

- Aquellos contenidos organizados cronológicamente se prestan a una indexación según fecha de publicación. Es frecuente combinar este factor con el de temática y el de selección de zonas del sitio web.

Selección desde los componentes de contenido hasta el índice

- Permitir al usuario la búsqueda según la tipología y características de los contenidos suele facilitar la recuperación y exploración del sitio web. Esto es según el formato y según los elementos de descripción que posean. Por ejemplo el cuerpo del texto, el título, la URL, el nombre del sitio, enlaces, enlaces a imágenes y a documentos, textos alternativos, descripciones, palabras clave, metadatos, meta-etiquetas, textos de anclaje, etc.
- En este sentido y combinado con las estrategias de búsqueda, se suelen crear salones de búsqueda o también más comúnmente denominadas páginas de búsqueda avanzada.

Los algoritmos de búsqueda que actúan dentro del buscador

¿Qué es un algoritmo de recuperación? Es un método diseñado para resolver un problema específico, en este caso, recuperar información a través de las consultas del usuario. Los algoritmos aparte de ser métodos de procedimiento y tratamiento de la información, así como de cálculo son instrumentos o herramientas que actúan conjuntamente con muchos otros mecanismos del motor de búsqueda. Por ejemplo, los indexadores, los webcrawlers, los visualizadores de contenidos, los algoritmos de inteligencia artificial para la interpretación de las consultas, los sistemas de gestión y ejecución temporal de los distintos algoritmos, etc. Todos ellos se articulan para formar lo que comúnmente se denomina buscador, y en general se puede denominar "en genérico" como algoritmo de recuperación. Los algoritmos más frecuentes para la búsqueda y localización de información se pueden clasificar en dos grandes categorías:

Algoritmos de reconocimiento de patrones

- Su objetivo es determinar la coincidencia de patrones entre la consulta del usuario y el corpus textual que ha sido indexado por el motor de búsqueda. Cuando se encuentran coincidencias, se recuperan los registros y se representan mostrando el patrón de consulta y las distintas coincidencias. A parte existe todo un aparataje de refinamiento para mejorar los resultados. Un ejemplo de este concepto son las expresiones regulares REGEXP o el reconocimiento de patrones LIKE en MySQL.

Algoritmos de reformulación y precisión

- Se basan en la relevancia de los resultados, su precisión y exhaustividad, devolviendo solamente los resultados de mayor calidad. Esto se consigue mediante la aplicación de métodos booleanos, vectoriales, de similaridad y probabilísticos.

Otros enfoques

- Cálculo del interés entre citas de un documento.
- Método de bibliografía activa, por relevancia y similaridad de los documentos referenciados.
- Documentación similar basada en el análisis de los términos del texto.

El futuro desarrollo de algoritmos

- Métodos de evaluación de consultas y documentos basados en la experiencia del usuario.

Generadores de consultas

Como ya se ha explicado el algoritmo de búsqueda propiamente dicho no es el único componente del buscador, de hecho tienen gran importancia las herramientas para ayudar a generar las consultas y asistir al usuario. A continuación se explicitan algunos de los métodos más comunes:

Auto-sugerencias

- Cuando se escribe de forma incorrecta un término el sistema sugiere el término correcto para que se repare la consulta y se proporcionen resultados pertinentes, garantizando una recuperación que por otro lado no devolvería probablemente ningún resultado.

Herramientas fonéticas

- Permiten determinar las equivalencias fonéticas de los términos de consulta para expandir la búsqueda y obtener resultados que a priori serían difíciles de recuperar.

Constructores de términos derivados

- A partir de la raíz de los términos que el usuario utiliza en la búsqueda es capaz de proponer al usuario otra consulta con nuevos términos basados en dicha raíz.

Herramientas de procesamiento del lenguaje natural

- Efectúan un análisis sintáctico de las oraciones, frases y sintagmas de las consultas del usuario para determinar la estrategia de búsqueda y localizar patrones más similares.

Vocabularios controlados

- Cuando el usuario introduce la cadena de consulta, se le sugieren términos controlados relacionados o parecidos a partir del tesoro o taxonomía que disponga integrada el sistema.

7. Representación de resultados e interfaz de búsqueda

El proceso de representación y visualización de resultados le sigue al proceso de recuperación de información en el que el buscador resuelve los documentos más pertinentes a la consulta del usuario. Las implicaciones que se derivan de la presentación de los resultados de la consulta son claves en arquitectura de la información. ¿Cómo representar la información?

Qué campos y tipos de datos deben ser visualizados

- Para los usuarios que conocen el resultado exacto, se muestran los datos esenciales, representativos de cada respuesta, como por ejemplo el autor y el título en cada registro.
- Para usuarios con mayor desconocimiento se recomiendan campos descriptivos como por ejemplo el resumen, las palabras clave, señalización de los términos en los párrafos.
- Cuando los campos con datos esenciales no ayudan a discernir correctamente se recomienda la introducción de otros, como por ejemplo la fecha o año de publicación, registro o fecha de actualización, localización, etc.

Cuántos documentos se deben mostrar

- Si la descripción de los documentos recuperados es amplia se recomienda mostrar menos resultados. En caso de ser más escueta se recomienda mostrar más resultados. También ello puede depender de terceros factores como por ejemplo la resolución para la que está optimizada la página de resultados o el navegador web que se esté empleando.
- Siempre que sea posible se deberán mostrar el número total de resultados obtenidos en cada consulta de forma tal que el usuario pueda valorar si realmente desea efectuar una navegación a través de todos ellos.
- Lo común es mostrar tandas de 10, 25, 50 o 100 resultados por página. También se deberá permitir su ajuste personalizado.

Listado de resultados

- Ordenación alfabética de los resultados. Los títulos de los resultados no alfabeticarán por artículos determinados, indeterminados o preposiciones.
- Ordenación cronológica de los resultados. Utilizado para aquellos resultados relacionados con la publicación de noticias o información periódica y continua.
- Ordenación de los resultados por relevancia. La relevancia de los resultados depende de la frecuencia de aparición u ocurrencias de los términos de consulta, presentes en el documento, de la proximidad de los términos o su adyacencia en la misma frase o párrafo, del campo en el que fue encontrado el término de la consulta (título, resumen, subtítulo o cuerpo del documento), de la popularidad del documento después de ser valorado por el usuario. Finalmente en este proceso también intervienen factores como la heterogeneidad de la colección (Cuanto más dispar, más dificultad para recuperar con relevancia) y el método de indexación de los textos y términos que componen el corpus documental.
- Ordenación de los resultados por popularidad. Implica un proceso de ponderación de los resultados en función de los enlaces externos que recibe cada recurso. Cuantos más enlaces referencian el recurso, más conocido y mejor valorado por los usuarios. Este tipo de algoritmos, de rango de páginas, tiene su ejemplo en el PageRank [<http://infolab.stanford.edu/~backrub/google.html>]
- Ordenación de los resultados según experiencia del usuario. El usuario se encarga de calificar los recursos en función de la experiencia y utilidad de los mismos para el desempeño de sus actividades, tales puntuaciones se suman y concretan en un valor numérico como por ejemplo número total de valoraciones y máxima calificación. Tales datos suelen estar acompañados por mecanismos de comentarios y etiquetado social.
- Ordenación por pago de emplazamiento. Cuando la aparición en las primeras posiciones de los rankings no se deben a ninguno de los motivos anteriormente expuestos, cabe la posibilidad de que el resultado fuera emplazado en esa posición mediante pago, se trata del factor económico. Aparecer entre los primeros resultados supone una mayor visibilidad y esto es aprovechado por muchos usuarios y empresas para promocionar sus servicios y productos. A este modelo se le conoce por sus siglas PFP (Pay For Placement)

Interfaz de búsqueda

El interfaz de búsqueda hace posible un lenguaje visual fácilmente comprensible para el usuario con el que poder emitir las órdenes y requerimientos de su consulta y necesidades informativas. Desde el punto de vista de la arquitectura de la información, el reto es presentar las opciones de búsqueda con un interfaz adaptado a cada situación, por ejemplo:

- **El nivel de experiencia del usuario y la motivación de la búsqueda.** Están familiarizados los usuarios con los lenguajes de interrogación y consulta como por ejemplo los operadores booleanos o prefieren emplear el lenguaje natural, necesitan ayudas contextuales para refinar las consultas, qué nivel de precisión necesitan para satisfacer sus necesidades, que nivel de exhaustividad requieren en los resultados, cuántas veces están dispuestos a reformular la consulta o hasta qué punto están dispuestos a navegar entre las páginas de resultados.
- **Tipo de necesidad de información.** Necesitan información para la toma de decisiones, qué tipo de detalles se necesitan proporcionar de cada documento o resultado para satisfacer esa demanda.
- **Tipo de información que se busca.** Se requiere una información estructurada por campos, tabulada o a texto completo, en qué formatos, con qué grado de actualización.
- **Cantidad de información que se busca.** Mostrar demasiada información puede abrumar al usuario, pero informarle en todo caso puede proporcionarle un alivio y una garantía de control en la representación y visualización de la información, por lo que dotar al interfaz de consulta de dicho control puede suponer.

En función de tales factores y casos, se diseña el interfaz de consulta o de control del motor de búsqueda, desde la estrategia de búsqueda por defecto, pasando por la que defina el usuario, hasta los campos, botones y selectores que posibilitan efectuar búsquedas booleanas, definir los formatos que se desean recuperar o el calendario que permite identificar las fechas extremas de la consulta. Todos estos elementos son los componentes de la interfaz de búsqueda.

Componentes de la interfaz de búsqueda

- **Caja de consulta.** Es el espacio desde el que se transmite la petición del usuario en forma de cadena de texto. Siempre está acompañada de un botón que emite la orden de consulta, desencadenando todo el proceso de búsqueda. El usuario suele dar por hecho que con utilizar los términos que describan su necesidad informativa, el motor de búsqueda hará el resto del trabajo, o que no es necesario introducir operadores booleanos, o que no hay que preocuparse por introducir términos ambiguos o sinónimos, o que no va a perder tiempo buscando más que por el campo principal de búsqueda (es decir la caja de texto), o bien que la consulta buscará en todos los documentos en su texto completo. Esta serie de actitudes, no educan ni enseñan nada al usuario y es por ello que después de la primera búsqueda es aconsejable recordar que tienen a su disposición métodos de refinamiento, sugerencias con otras cadenas de consulta correctamente escritas, siendo ese momento en el que aprenden y reformulan la consulta (Que es uno de los objetivos cuando no se encuentra la información directamente). Con cada movimiento y reformulación el sistema es capaz de conocer mejor el objeto de la consulta y probablemente pueda ser mejor refinado. En cuanto al posicionamiento de la caja de consulta, puede variar dependiendo del tipo de sitio web de que se trate. Por ejemplo un buscador lo situará en lugar preferentemente central, pero un sitio web con otro tipo de contenidos tenderá a situarlo en la misma barra de navegación global o dentro del menú contextual.
- **Búsqueda avanzada.** Consiste en una página con funcionalidades de control del motor de búsqueda, en las que un usuario especializado o no satisfecho con el modo de consulta básico puede refinar una consulta para encontrar información más precisa. Aunque el porcentaje de usuarios potenciales de este componente es altísimo, el porcentaje que realmente utiliza la búsqueda avanzada es bajo. La tendencia es proporcionar junto con los resultados de la primera búsqueda las opciones de refinamiento propias de la búsqueda avanzada. En todo caso la forma de la búsqueda avanzada es un conjunto de campos o cajas de consulta para especificar términos obligatorios, no presentes, de relevancia, especificación geográfica, idiomática, búsqueda por frase exacta, especificación del rango de fechas, etc.

- **Repetición de consulta original.** Se trata de mostrar al usuario dentro de una capa o área concreta de la página de resultados, su propia consulta. El objeto de ello es evitar que olvide los términos de su consulta y permitir la reformulación de su consulta.
- **Cuando todo falla.** Qué ocurre si el sistema no proporciona ningún resultado al usuario. Sería recomendable mostrarle sugerencias de consulta que lo redireccionen a terceros resultados con el fin de que encuentre de nuevo el camino, vestigio o pista de su necesidad informativa. Pero también puede ocurrir que todo falle cuando se recuperan miles o millones de resultados y el usuario necesite reducir notablemente su número, es decir, encontrar dentro del pajar la aguja. La forma de ayudar a su resolución es revisar la búsqueda, asesorar al usuario sobre cómo refinar, enfocar su reformulación con la navegación de la página, o contactar con un asistente para su consulta.

8. Usabilidad web

El concepto Usabilidad como tal, no está recogido en el Diccionario de la Real Academia, debe su origen, probablemente, a alguno de los siguientes términos en inglés *Usability = Usabilidad*, *User friendly = Fácil de usar*, *Friendly use = Facilidad de uso*, véase (HASSAN, Y.; MARTÍN FERNÁNDEZ, F.J.; GHZALA, I. 2004). Según su origen, se advierte que la facilidad de uso es el objetivo de la usabilidad, que adaptado al caso de la web, implica un uso sencillo y amigable de la misma.

Según la norma ISO 9126, usabilidad se define como *"la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso"*. De forma muy parecida también es definida por la norma ISO 9241 aunque con un enfoque más propio de la mercadotecnia *"eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico"*.

Otro punto de vista más especializado (NIELSEN, J. 2006) entiende que la usabilidad web es *"la aclaración del propósito del sitio, ayudar a los usuarios a encontrar lo que buscan, mostrar el contenido del sitio convenientemente y diseñar el sitio web para mejorar la interacción con el usuario, evitando ser definida a priori"*.

Como elemento divulgativo añadido, se puede proponer una definición en la que se aclare explícitamente que la experiencia y la inteligencia del diseño son, al sentido común con el que se diseñan las páginas web, la mejor usabilidad posible:

"Usabilidad web es la simplificación necesaria de las funciones y contenidos de un sitio o página web en sus elementos denotativos y connotativos esenciales, de forma tal que su motivación, objeto, navegación y contenidos estén al alcance del usuario de forma clara y sencilla, visualmente y cognitivamente identificables, recuperables con inmediatez, expresados y concebidos con la inteligencia del diseñador y la experiencia del usuario". BLÁZQUEZ OCHANDO, M.

El concepto de usabilidad según Steve Krug

Según Steve Krug, la usabilidad consiste en evitar que el usuario del sitio web tenga que pensar cómo acceder a los contenidos, con qué elementos debe interactuar o que se pregunte sobre el significado de las opciones, la navegación, cómo recuperar y en definitiva cómo está enfocado o diseñado el sistema. Por lo tanto la regla central de la usabilidad, consiste en evitar que el usuario se pregunte dónde se encuentra cada cosa.

- **Qué nos hace pensar.** Si el objetivo es lograr que el usuario piense menos, se debe procurar que los botones realmente parezcan botones, los enlaces sean enlaces de verdad y los términos que se utilicen sean concisos y breves. Un usuario pensará o reflexionará más si en el sitio web, encuentra sintagmas o frases en los enlaces, si los botones tienen una forma parecida a un botón pero no lo son o si los enlaces tienen flechas o aditivos visuales como flechas, corchetes, etc.
- **Cómo se utiliza la web.** Uno de los comportamientos más habituales consiste en ojear las páginas que componen un sitio web. Este proceso indica que en la mayoría de las ocasiones el usuario no lee completamente todo el contenido textual de las páginas, centrandose su actividad reflexiva en unos pocos detalles que tendrán que ser suficientemente denotativos como para tomar una decisión y continuar en la misma página o acceder a la siguiente. Durante este proceso que podría denominarse "echar un vistazo" dichos elementos denotativos se convierten en lo único importante para el usuario, el resto de detalles, textos, gráficos y opciones dejan de existir.
- Este comportamiento puede deberse a la necesidad de resolver rápidamente las necesidades y problemas que al usuario se le plantean, por motivo de economizar el tiempo que tiene disponible ó reducir su frustración cuando visita una página web y no encuentra directamente lo que estaba buscando. Otra razón por la que "echar una ojeada" resulta común es que sencillamente no se necesita leer todo el contenido para comprenderlo, de hecho este acto casi reflejo, se ha convertido en una costumbre después de que se esté habituado a repasar periódicos, libros, revistas, exportando el comportamiento a la web.

- Cuando el usuario ojea, no siempre elige la mejor opción para llegar a la información que necesita. No siempre es la decisión óptima, pero sí la más razonable. Ello sucede porque no hay consecuencias graves si el usuario se equivoca en la página a cargar. También es posible que el usuario desee conocer otras opciones o alternativas para descubrir qué contenidos que pueden ser de utilidad y descartar para futuras ocasiones cuáles no. Todo ello hace que la combinación entre navegación, búsqueda y ojeo sea un trabajo creativo e intuitivo para el usuario y que se genere una experiencia de uso que posteriormente se tratará de aprovechar.
- En conclusión se puede afirmar que sin grandes explicaciones e instrucciones, el usuario es capaz de hacerse una *"composición de lugar"*, *"una idea"*, *"un bosquejo"* sobre cómo funciona un sitio web, ojeando y navegando entre sus páginas, acertando en mayor o menor medida con respecto a la idea original de su diseñador.

Consideraciones de diseño

- **Jerarquía visual.** Lo más importante deberá ser destacado del resto. Los contenidos similares tendrán que tener un estilo de identificación visual y tipográfica homogéneo a lo largo de todo el sitio web. Los contenidos referentes a un titular o epígrafe de nivel superior, deberán estar identificados en la misma área para que sea evidente su dependencia.
- **Dónde se hace clic.** Dejar claro qué es un enlace y qué no lo es, puede ayudar a reducir el número de clics a lo largo del sitio web. Se estima que 3 clics equivalen a 1 clic claro y reflexionado. Cuando la navegación impide el acceso directo, se necesita conseguir redireccionar al usuario entre distintas opciones, pero por regla general un sitio web más horizontal, será más fácil de acceder que uno diseñado con muchos niveles verticales en su jerarquía.
- **Eliminar lo accesorio.** Esto es omitir palabras innecesarias, que no expresen de forma precisa el objeto, la significación, y el hilo argumental de los contenidos.

Realzar el contenido práctico y esencial para conseguir páginas más cortas y visibles con un solo golpe de vista.

- **Minimizar el ruido visual.** Implica reducir los efectos visuales excesivos y molestos que alerten constantemente la atención del usuario y que perjudiquen la máxima claridad de los contenidos y de las opciones de navegación del sitio web. Estos elementos son líneas de puntos, excesivos subrayados, imágenes de fondo repetidas, imágenes demasiado grandes, abundantes efectos “*rollovers*”, excesivo aparataje de interfaz que sobrecarga la claridad del conjunto.

- **Diseño de la navegación.** Utilizar sistemas de navegación como Breadcrumbs, definir rutas de navegación a todos los contenidos del sitio mediante pocos niveles jerárquicos, en beneficio de sitios web más horizontales y con una mejor estructuración de secciones. El objeto de la navegación es que el usuario encuentre lo que necesita, motivo por el cual tiene que tener libertad para elegir, ya sea utilizando el buscador, consultando las preguntas frecuentes u ojeando por su cuenta. Por otra parte la navegación debe ser un elemento clarificador que indique en qué lugar se encuentra el usuario visualizando una página, de qué sección procede, qué otras están clasificadas de la misma forma, que proporcione un método de acceso a los contenidos del sitio que sea extrapolable a otros muchos casos y que permita establecer un cierto grado de confianza con respecto a la correcta organización del resto de contenidos.

- **Convenciones.** Definir en todo momento en qué página se encuentra el usuario mediante Breadcrumbs o “usted está aquí”. Mantener siempre visibles los elementos de navegación como los Breadcrumbs. Utilizar secciones y subsecciones ordenadas jerárquicamente. Emplear navegación local con una columna y su correspondiente lista de enlaces. Opcionalmente usar un menú de tipo sólo texto en el pie de página con los principales accesos directos del sitio web.

Evaluación de la usabilidad

- Qué sitio es (Identificar el sitio web)
- En qué página estoy (Nombre página)
- Cuáles son las principales secciones del sitio (secciones)
- Qué opción tengo en este nivel (Navegación local)
- Dónde estoy en el esquema de cosas (Usted está aquí + Breadcrumbs)
- Cómo busco algo (Buscador)

9. Diseño de estructuras web con capas y CSS: layouts

Para diseñar la estructura de una página web, existen diversos métodos:

- **Uso de marcos o frames** para separar los distintos espacios del sitio web. Implica el uso de las etiquetas “<frameset>”, véase tabla 1. Este modelo muy popular en los primeros desarrollos de la web, posteriormente ha sido desechado debido a las dificultades de accesibilidad que presentaba y a los problemas de redimensionamiento y correcta visualización en los navegadores. No es aconsejable utilizar este modelo.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>Estructura con frames</title>
</head>
<frameset cols="30%, 70%">
  <frameset rows="100, 200">
    <frame src="contenido1.html">
    <frame src="contenido2.html">
  </frameset>
<noframe>
<ul>
<li><a href="content1.html">contenido primero</a></li>
<li><a href="content2.html">contenido segundo</a></li>
</ul>
</noframe>
</frameset>
</html>
```

Tabla 1. Estructura web basada en marcos o frames

Uso de capas “<div>” con estilos CSS para definir distintas áreas, se puede considerar el método de mayor éxito y que mejor resuelve los problemas de accesibilidad y estructuración de un sitio web, véase la siguiente tabla. La estructura puede ser perfectamente redimensionable y totalmente adaptable a cualquier caso. El estilo definido siempre es interpretado por todos los navegadores web.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Estructura con capas y CSS</title>

<style>
.capa1 {width: 30%;
```

```
height: 500px;
float: left;
background-color: #800000;

.capa2 {width: 70%;
height: 500px;
float: left;
background-color: #273a53;}
</style>

</head>
<body>

<div class='capa1'></div>
<div class='capa2'></div>

</body>
</html>
```

Tabla 2. Estructura web basada en capas

Uso de “<iframe>” y capas “<div>”. Puede ocurrir que se utilice un método mixto para estructurar los contenidos. Los iframe son marcos parecidos a los frames, con la diferencia de que no dividen la pantalla vertical u horizontalmente. Lo que se obtiene es una ventana dentro de la capa “<div>” que la soporta, permitiendo en todo caso un mayor control de accesibilidad y redimensionamiento que el primer caso expuesto. Otras ventajas derivadas es la posibilidad de cargar contenidos direccionados desde los enlaces de los distintos menús de forma sencilla, véase siguiente tabla.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>

<style>
.capa1 {width: 100%;
height: 150px;
clear: both;
background-color: #364d35;}

.capa2 {width: 30%;
height: 500px;
float: left;
background-color: #800000;}

.capa3 {width: 70%;
height: 500px;
float: left;
background-color: #273a53;}
</style>

</head>
<body>

<div class='capa1'>Encabezamiento</div>
<div class='capa2'>
<ul>
<li><a href='contenido1.html' target='hud'>enlace1</a></li>
<li><a href='contenido2.html' target='hud'>enlace2</a></li>
<li><a href='contenido3.html' target='hud'>enlace3</a></li>
</ul>
```

```
</div>

<div class='capa3'>
<iframe src='contenido0.html' name='hud' id='hud'></iframe>
</div>


</body>
</html>
```

Tabla 3. Estructura web basada en capas e iframe

Estructuras o disposiciones de sitios web

Modelo 1

<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/layout-css-01.html>



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>

<style>
.capa1 {width: 100%; height: 50px; clear: both; background-color: #b3c5d1;}
.capa2 {width: 30%; height: 200px; float: left; background-color: #d1e0eb;}
.capa3 {width: 70%; height: 200px; float: left; background-color: #eef0f0;}
.capa4 {width: 100%; height: 50px; clear: both; background-color: #b6bbbe;}
.contenido {padding: 10px;}
</style>

</head>
<body>

<div class='capa1'>
<div class='contenido'>texto1</div>
</div>

<div class='capa2'>
<div class='contenido'>texto2</div>
</div>


<div class='capa3'>
<div class='contenido'>texto3</div>
</div>

<div class='capa4'>
<div class='contenido'>texto4</div>
</div>

</body>
</html>
```

Modelo 2

<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/layout-css-02.html>



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>

    <style>
    .capa1 {width: 100%; height: 50px; clear: both; background-color: #b3c5d1;}
    .capa2 {width: 20%; height: 200px; float: left; background-color: #d1e0eb;}
    .capa3 {width: 60%; height: 200px; float: left; background-color: #eef0f0;}
    .capa4 {width: 20%; height: 200px; float: left; background-color: #d3ebf1;}
    .capa5 {width: 100%; height: 50px; clear: both; background-color: #b6bbbe;}
    .contenido {padding: 10px;}
    </style>

</head>
<body>

<div class='capa1'>
<div class='contenido'>texto1</div>
</div>

<div class='capa2'>
<div class='contenido'>texto2</div>
</div>

<div class='capa3'>
<div class='contenido'>texto3</div>
</div>


<div class='capa4'>
<div class='contenido'>texto4</div>
</div>

<div class='capa5'>
<div class='contenido'>texto5</div>
</div>

</body>
</html>
```

Modelo 3

<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/layout-css-03.html>



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>

    <style>
        .capa1 {width: 100%; height: 50px; clear: both; background-color: #b3c5d1;}
        .capa2 {width: 20%; height: 200px; float: left; background-color: #d1e0eb;}
        .capa3 {width: 20%; height: 200px; float: left; background-color: #eef0f0;}
        .capa4 {width: 20%; height: 200px; float: left; background-color: #e1eaed;}
        .capa5 {width: 40%; height: 200px; float: left; background-color: #cbdde5;}
        .capa6 {width: 100%; height: 50px; clear: both; background-color: #b6bbbe;}
        .contenido {padding: 10px;}
    </style>

</head>
<body>

<div class='capa1'>
<div class='contenido'>texto1</div>
</div>

<div class='capa2'>
<div class='contenido'>texto2</div>
</div>

<div class='capa3'>
<div class='contenido'>texto3</div>
</div>

<div class='capa4'>
<div class='contenido'>texto4</div>
</div>

<div class='capa5'>
<div class='contenido'>texto5</div>
</div>

<div class='capa6'>
<div class='contenido'>texto6</div>
</div>

</body>
</html>
```

Modelo 4

<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/layout-css-04.html>



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>
```

```
<style>
.capa1 {width: 100%; height: 50px; background-color: #b3c5d1;}
.capa2 {width: 100%; height: 50px; background-color: #eef0f0;}
.capa3 {width: 100%; height: 200px; float: left; background-color: #d1e0eb;}
.capa4 {width: 100%; height: 50px; clear: both; background-color: #b6bbbe;}
.contenido {padding: 10px;}
</style>

</head>
<body>

<div class='capa1'>
<div class='contenido'>texto1</div>
</div>

<div class='capa2'>
<div class='contenido'>texto2</div>
</div>

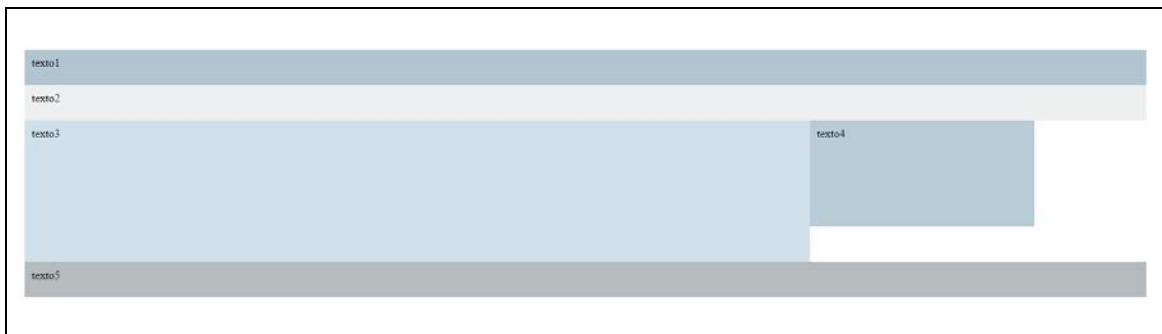
<div class='capa3'>
<div class='contenido'>texto3</div>
</div>

<div class='capa4'>
<div class='contenido'>texto4</div>
</div>

</body>
</html>
```

Modelo 5

<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/layout-css-05.html>



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>

<style>
.capa1 {width: 100%; height: 50px; background-color: #b3c5d1;}
.capa2 {width: 100%; height: 50px; background-color: #eef0f0;}
.capa3 {width: 70%; height: 200px; float: left; background-color: #d1e0eb;}
.capa4 {width: 20%; height: 150px; float: left; background-color: #baccd6;}
.capa5 {width: 100%; height: 50px; clear: both; background-color: #b6bbbe;}
.contenido {padding: 10px;}
</style>

</head>
<body>

<div class='capa1'>
<div class='contenido'>texto1</div>
</div>

<div class='capa2'>
<div class='contenido'>texto2</div>
```

```
</div>

<div class='capa3'>
<div class='contenido'>texto3</div>
</div>

<div class='capa4'>
<div class='contenido'>texto4</div>
</div>

<div class='capa5'>
<div class='contenido'>texto5</div>
</div>

</body>
</html>
```

Modelo 6

<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/layout-css-06.html>



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>

<style>
.supercapa {width: 80%; margin: 0px auto; padding: 0px;}
.capa1 {width: 100%; height: 50px; background-color: #b3c5d1;}
.capa2 {width: 100%; height: 50px; background-color: #eef0f0;}
.capa3 {width: 70%; height: 200px; float: left; background-color: #d1e0eb;}
.capa4 {width: 30%; height: 150px; float: left; background-color: #baccd6;}
.capa5 {width: 100%; height: 50px; clear: both; background-color: #b6bbb6;}
.contenido {padding: 10px;}
</style>

</head>
<body>

<div class='supercapa'>

<div class='capa1'>
<div class='contenido'>texto1</div>
</div>

<div class='capa2'>
<div class='contenido'>texto2</div>
</div>

<div class='capa3'>
<div class='contenido'>texto3</div>
</div>

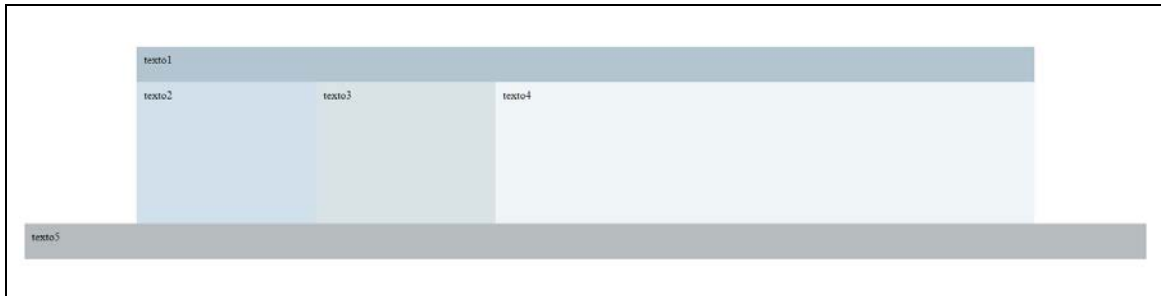
<div class='capa4'>
<div class='contenido'>texto4</div>
</div>

<div class='capa5'>
<div class='contenido'>texto5</div>
</div>
```

```
</div>  
</body>  
</html>
```

Modelo 7

<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/layout-css-07.html>



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html>  
<head>  
<title></title>  
  
  <style>  
    .supercapa {width: 80%; margin: 0px auto; padding: 0px;}  
    .capa1 {width: 100%; height: 50px; background-color: #b3c5d1;}  
    .capa2 {width: 20%; height: 200px; float: left; background-color: #d1e0eb;}  
    .capa3 {width: 20%; height: 200px; float: left; background-color: #d9e3e7;}  
    .capa4 {width: 60%; height: 200px; float: left; background-color: #eff5f7;}  
    .capa5 {width: 100%; height: 50px; clear: both; background-color: #b6bbbe;}  
    .contenido {padding: 10px;}  
  </style>  
  
</head>  
<body>  
  
<div class='supercapa'>  
  
  <div class='capa1'>  
    <div class='contenido'>texto1</div>  
  </div>  
  
  <div class='capa2'>  
    <div class='contenido'>texto2</div>  
  </div>  
  
  <div class='capa3'>  
    <div class='contenido'>texto3</div>  
  </div>  
  
  <div class='capa4'>  
    <div class='contenido'>texto4</div>  
  </div>  
  
</div>  
  
<div class='capa5'>  
<div class='contenido'>texto5</div>  
</div>  
  
</body>  
</html>
```

10. Estilo de enlaces, menús y vinculación de archivos CSS

Enlaces

Los enlaces constituyen la base de la Hipertextualidad en el sitio web. Desde el punto de vista de la arquitectura de la información deben proporcionar toda la información que sea posible en relación a su título o descripción, idioma de la página de destino y marco o página en la que será cargado el enlace. Ello se consigue con la sintaxis que se muestra en la tabla 4.

```
<a href='dirección URL del enlace' target='página de destino' hreflang='código de idioma ISO 639-1' title='título o descripción del enlace' rel='relación entre el documento y el recurso enlazado'>nombre de enlace</a>
```

Tabla 4. Enlace en HTML con todos sus atributos

Para asignar estilos a los enlaces, se necesita utilizar CSS y señalar las propiedades para los 4 posibles estados en los que se puede encontrar dicho enlace: 1) “*a:link*” el enlace en estado normal; 2) “*a:visited*” el enlace fue visitado por el usuario; 3) “*a:hover*” cuando el cursor se mueve por encima del enlace; 4) “*a:active*” cuando el usuario presiona sobre el enlace sin soltar el botón del ratón. Todos estos estados se pueden definir, tal y como se muestra en la tabla 5.

```
a:link { color: #224a6a; font-size: 12px; font-family: arial; }
a:visited { color: #236b6a; font-size: 12px; font-family: arial; }
a:hover { color: #4497da; font-size: 12px; font-family: arial; }
a:active { color: #44c5da; font-size: 12px; font-family: arial; }
```

Tabla 5. Estilo CSS para los enlaces HTML

El resultado de aplicar los estilos de la tabla 5 a una lista de enlaces es el ejemplo presentado en el modelo1 de enlaces, que guarda todas las reglas y recomendaciones propuestas para la arquitectura de la información y su mayor accesibilidad.

Modelo 1 de enlaces

<http://www.mblazquez.es/blog-ccdodoc-arquitectura-informacion/documentos/links-css-01.html>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

<pre> <html> <head> <title></title> <style> a:link {color: #224a6a; font-size: 20px; font-family: arial, sans-serif;} a:visited {color: #236b6a; font-size: 20px; font-family: arial, sans-serif;} a:hover {color: #4497da; font-size: 20px; font-family: arial, sans-serif;} a:active {color: #44c5da; font-size: 20px; font-family: arial, sans-serif;} </style> </head> <body> <div> Google Yahoo Exalead </div> </body> </html> </pre>
<ol style="list-style-type: none"> 1. Google 2. Yahoo 3. Exalead

Tabla 6. Modelo1 de enlaces

Pero puede suceder el caso de que se necesiten distintos estilos de enlaces en la misma página. El problema de diferenciar distintos estilos en CSS y aplicarlos en los enlaces se resuelve, añadiendo un identificador para cada tipo de enlace, véase tabla 7.

<pre> a:link.enlace1 { color: #224a6a; font-size: 12px; font-family: arial; } a:visited.enlace1 { color: #236b6a; font-size: 12px; font-family: arial; } a:hover.enlace1 { color: #4497da; font-size: 12px; font-family: arial; } a:active.enlace1 { color: #44c5da; font-size: 12px; font-family: arial; } </pre>
--

Tabla 7. Diferenciando dos estilos de enlaces

De esta forma, véase modelo2 de enlaces, se consigue crear tantos estilos de enlaces como fueren necesarios para una o muchas páginas distintas. Este método resulta de gran importancia cuando se diseñan menús con CSS, ya que a menudo en una misma página conviven múltiples estilos que atañen a muy diversas formas gráficas que adoptan los enlaces.

Modelo 2 de enlaces

<http://www.mblazquez.es/blog-ccdodoc-arquitectura-informacion/documentos/links-css-02.html>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>
<style>
a:link.enlace1 {color: #224a6a; font-size: 20px; font-family: arial, sans-serif;}
a:visited.enlace1 {color: #236b6a; font-size: 20px; font-family: arial, sans-serif;}
a:hover.enlace1 {color: #4497da; font-size: 20px; font-family: arial, sans-serif;}
a:active.enlace1 {color: #44c5da; font-size: 20px; font-family: arial, sans-serif;}
a:link.enlace2 {color: #790d0e; font-size: 20px; font-family: arial, sans-serif;}
a:visited.enlace2 {color: #79230d; font-size: 20px; font-family: arial, sans-serif;}
a:hover.enlace2 {color: #d12f27; font-size: 20px; font-family: arial, sans-serif;}
a:active.enlace2 {color: #d13b27; font-size: 20px; font-family: arial, sans-serif;}
</style>
</head>
<body>
<div style='font-size: 24px; font-family: arial, sans-serif;'>Tanda de enlaces1:
Buscadores</div>
<div>
<ol>
<li><a href='http://www.google.es' class='enlace1' target='_blank' hreflang='es'
title='Buscador Google'>Google</a></li>
<li><a href='http://es.yahoo.com/' class='enlace1' target='_blank' hreflang='es'
title='Buscador Yahoo'>Yahoo</a></li>
<li><a href='http://www.exalead.com/search/' class='enlace1' target='_blank' hreflang='en'
title='Buscador Exalead'>Exalead</a></li>
</ol>
</div>
</div>
<div style='font-size: 24px; font-family: arial, sans-serif;'>Tanda de enlaces2:
Directorios</div>
<div>
<ol>
<li><a href='http://www.webworldindex.com/' class='enlace2' target='_blank' hreflang='en'
title='Directorio Web World'>Web World</a></li>
<li><a href='http://dir.yahoo.com/' class='enlace2' target='_blank' hreflang='en'
title='Directorio de Yahoo'>Yahoo Directory</a></li>
<li><a href='http://www.dmoz.org/' class='enlace2' target='_blank' hreflang='en'
title='Directorio DMOZ'>DMOZ Directory</a></li>
</ol>
</div>
</div>
</body>
</html>
```

Tanda de enlaces1: Buscadores

1. [Google](#)
2. [Yahoo](#)
3. [Exalead](#)

Tanda de enlaces2: Directorios

1. [Web World](#)
2. [Yahoo Directory](#)
3. [DMOZ Directory](#)



Tabla 8. Modelo 2 de enlaces

Menús

Desde los comienzos de los primeros sitios web hasta la fecha una de las constantes que más preocupaciones causaron fue el asunto de los menús. Se han diseñado complejos menús en Java, Applets, sofisticados menús en Flash, Shockwave, pero en ninguno de los casos, resultaban accesibles, sencillos y suficientemente bien estructurados como para ser indexados con facilidad por cualquier webcrawler. Por ello en arquitectura de la información se aboga por un diseño lo más sencillo posible que favorezca una fácil lectura, con contraste adecuado, una correcta visibilidad, gran limpieza, simplicidad, funcionalidad y una coloración agradable en tonos *fríos-mates-nobrillantes-opacos-escalares-gradientes-suaves-pasteles-grisaceos-blanquecinos-claros*.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>
<style>
.menuhor {padding: 5px; background-color: #436781; border-radius: 15px; text-align: center;}
.contenedor {margin: 5px; padding: 5px;}
a:link.mlink {margin: 5px; padding: 10px; background-color: #dee4e6; border-radius: 15px;
color: #224a6a; font-size: 20px; font-family: arial, sans-serif;}
a:visited.mlink {margin: 5px; padding: 10px; background-color: #dee4e6; border-radius: 15px;
color: #236b6a; font-size: 20px; font-family: arial, sans-serif;}
a:hover.mlink {margin: 5px; padding: 10px; background-color: #f6f8f8; border-radius: 15px;
color: #4497da; font-size: 20px; font-family: arial, sans-serif;}
a:active.mlink {margin: 5px; padding: 10px; background-color: #f6f8f8; border-radius: 15px;
color: #44c5da; font-size: 20px; font-family: arial, sans-serif;}
</style>
</head>
<body>
<div class='menuhor'>
<div class='contenedor'>
<a href='#' class='mlink'>opcion1</a>
<a href='#' class='mlink'>opcion2</a>
<a href='#' class='mlink'>opcion3</a>
```

```

<a href='#' class='mlink'>opcion4</a>
<a href='#' class='mlink'>opcion5</a>
</div>
</div>
</body>
</html>

```



Tabla 9. Modelo de menú horizontal

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>
<style>
.supermenu { width: 100%; height: 56px; background-color: #f3f3f3; border: solid 1px #cecece;
border-radius: 5px; }
.elemento { float: left; }
ul.menu { position: relative; list-style: none; }
ul.menu ul { display: none; list-style: none; margin-top: 10px; border: solid 1px #bbbbbb;
background-color: #dbdbdb; border-radius: 5px; }
ul.menu li:hover > ul { position: absolute; display: block; width: 100px; line-height: 30pt;
text-indent: -20px; }
a:link.moplink { font-family: arial; font-size: 12pt; color: #ffffff; padding: 10px; text-
decoration: none; background-color: #dbdbdb; border-radius: 5px; }
a:visited.moplink { font-family: arial; font-size: 12pt; color: #ffffff; padding: 10px; text-
decoration: none; background-color: #dbdbdb; border-radius: 5px; }
a:hover.moplink { font-family: arial; font-size: 12pt; color: #ffffff; padding: 10px; text-
decoration: underline; background-color: #b1b1b1; border-radius: 5px; }
a:active.moplink { font-family: arial; font-size: 12pt; color: #ffffff; padding: 10px; text-
decoration: underline; background-color: #b1b1b1; border-radius: 5px; }
a:link.mlink { font-family: arial; font-size: 14pt; color: #ffffff; padding: 10px; text-
decoration: none; background-color: #dbdbdb; border: solid 1px #cecece; border-radius: 5px; }
a:visited.mlink { font-family: arial; font-size: 14pt; color: #ffffff; padding: 10px; text-
decoration: none; background-color: #dbdbdb; border: solid 1px #cecece; border-radius: 5px; }
a:hover.mlink { font-family: arial; font-size: 14pt; color: #ffffff; padding: 10px; text-
decoration: underline; background-color: #b1b1b1; border: solid 1px #cecece; border-radius:
5px; }
a:active.mlink { font-family: arial; font-size: 14pt; color: #ffffff; padding: 10px; text-
decoration: underline; background-color: #b1b1b1; border: solid 1px #cecece; border-radius:
5px; }
</style>
</head>
<body>
<div class='supermenu'>
<div class='elemento'>
<ul class='menu'>
<li><a href='#' class='mlink'>Categoria1</a>
<ul>
<div style='width: 100%; height: 10px;'></div>
<li><a class='moplink' href='#'>Opcion1</a></li>
<li><a class='moplink' href='#'>Opcion2</a></li>
<li><a class='moplink' href='#'>Opcion3</a></li>
<li><a class='moplink' href='#'>Opcion4</a></li>

```

```

<li><a class='moplink' href='#'>Opcion5</a></li>
<div style='width: 100%; height: 10px;'></div>
</ul>
</li>
</ul>
</div>

<div class='elemento'>
<ul class='menu'>
<li><a href='#' class='mlink'>Categoría2</a>
<ul>
<div style='width: 100%; height: 10px;'></div>
<li><a class='moplink' href='#'>Opción1</a></li>
<li><a class='moplink' href='#'>Opción2</a></li>
<li><a class='moplink' href='#'>Opción3</a></li>
<li><a class='moplink' href='#'>Opción4</a></li>
<li><a class='moplink' href='#'>Opción5</a></li>
<div style='width: 100%; height: 10px;'></div>
</ul>
</li>
</ul>
</div>

<div class='elemento'>
<ul class='menu'>
<li><a href='#' class='mlink'>Categoría3</a>
<ul>
<div style='width: 100%; height: 10px;'></div>
<li><a class='moplink' href='#'>Opción1</a></li>
<li><a class='moplink' href='#'>Opción2</a></li>
<li><a class='moplink' href='#'>Opción3</a></li>
<li><a class='moplink' href='#'>Opción4</a></li>
<li><a class='moplink' href='#'>Opción5</a></li>
<div style='width: 100%; height: 10px;'></div>
</ul>
</li>
</ul>
</div>
</div>
</body>
</html>

```



Tabla 10. Modelo de menú desplegable vertical

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>

<style>
.menuver { width: 150px; background-color: #b8c2c4; border-radius: 15px; text-align: center;}
.elemento { margin-top: 25px; margin-bottom: 25px; }
a:link.mlink { padding: 10px; background-color: #dee4e6; border-radius: 15px; color: #224a6a;
font-size: 20px; font-family: arial, sans-serif; text-decoration: none;}
a:visited.mlink { padding: 10px; background-color: #dee4e6; border-radius: 15px; color:
#236b6a; font-size: 20px; font-family: arial, sans-serif; text-decoration: none;}
a:hover.mlink { padding: 10px; background-color: #f6f8f8; border-radius: 15px; color: #4497da;
font-size: 20px; font-family: arial, sans-serif;}
a:active.mlink { padding: 10px; background-color: #f6f8f8; border-radius: 15px; color:
#44c5da; font-size: 20px; font-family: arial, sans-serif;}
</style>
</head>
<body>

<div class='menuver'>
<div style='height: 10px;'></div>
<div class='elemento'><a href='#' class='mlink'>&#8212; opcion1</a></div>
<div class='elemento'><a href='#' class='mlink'>&#8212; opcion2</a></div>
<div class='elemento'><a href='#' class='mlink'>&#8212; opcion3</a></div>
<div class='elemento'><a href='#' class='mlink'>&#8212; opcion4</a></div>
<div class='elemento'><a href='#' class='mlink'>&#8212; opcion5</a></div>
<div style='height: 10px;'></div>
</div>
</body>
</html>

```



Tabla 11. Modelo de menú lateral

Vinculación de archivos CSS

En todos los ejemplos vistos hasta el momento, los códigos de estilo CSS están integrados en las páginas HTML. A modo de ejemplo es muy útil dicha disposición, pero en la vida real, lo normal es referenciar en el documento HTML el archivo CSS que contiene todos los estilos que se utilizarán a lo largo del sitio web. El código para su referenciación es el que se muestra en la tabla 12. Como se puede observar no

solamente es posible enlazar un único archivo, pueden agregarse tantos como fueren necesarios. Ello dependerá del tipo de estilos o el uso de los estilos en cada archivo CSS. Por ejemplo un `estilo1.css` puede encargarse de la maquetación general del sitio. El archivo `estilo2.css` se dedica a la presentación de textos y contenidos publicados y finalmente un archivo `estilo3.css` para las distintas propiedades de los menús del sitio web.

```
<html>
<head>
<title></title>
<link rel='stylesheet' type='text/css' href='css/estilo1.css' />
<link rel='stylesheet' type='text/css' href='css/estilo2.css' />
<link rel='stylesheet' type='text/css' href='css/estilo3.css' />
</head>
<body>
</body>
</html>
```

Tabla 12. Vinculación de archivos CSS

11. Estructura de carpetas y archivos

La correcta estructuración de las carpetas y archivos del sitio web es un factor clave para su buena comprensión y gestión por parte de su administrador, desarrollador y arquitecto de la información. Existe una serie de carpetas y archivos cuyas denominaciones no deberían variar, ya que tener unas convenciones básicas es el principio de toda normalización. La estructura de un sitio web estático podría ser similar a la que se presenta en la figura 4.







Estructura típica de una web estática		
	css	Almacena todos los archivos de estilo CSS.
	documents	Se ubican todos los tipos de documentos, archivos y páginas HTML.
	images	Se guardan todas las imágenes de tipo ilustrativo que acompañan al contenido.
	interface	Se guardan todos los gráficos utilizados en el interfaz visual del sitio web.
	java	Carpeta en la que se guardan los archivos applet de java.
	index.html	Página principal e inicial del sitio web a partir de la cual se accede a todos los contenidos.

Figura 4. Muestra de estructura básica de un sitio web estático

En cambio cuando se desarrolla un sitio web dinámico, especialmente si se codifica con lenguaje PHP o similar, la organización de los contenidos puede variar notablemente. Por ejemplo se crean carpetas específicas para las funciones básicas del sitio web, para los módulos y extensiones que se añadan así como para los archivos de instalación. El concepto es completamente diferente incluso en el caso de los archivos, por ejemplo los archivos contenidos en la carpeta “*functions*”, deberían tener un prefijo “*func.nombre.php*” para especificar que pertenecen a dicha carpeta. Lo mismo ocurre con los módulos con el prefijo “*mod.nombre.php*”, véase figura 5.

Estructura típica de una web dinámica



css

Almacena todos los archivos de estilo CSS.



documents

Se ubican todos los tipos de documentos, archivos y páginas HTML.



images

Se guardan todas las imágenes de tipo ilustrativo que acompañan al contenido.



interface

Se guardan todos los gráficos utilizados en el interfaz visual del sitio web.



java

Carpeta en la que se guardan los archivos applet de java.



functions

Carpeta que almacena las distintas funciones básicas del sitio web, como por ejemplo páginas de resultados, páginas con interfaz de consulta, funciones de "login" para los usuarios, funciones de edición de artículos y publicaciones, gestión de contenidos del sitio web, etc. Los archivos contenidos deberían tener la forma "func.nombre.php", es decir, prefijo identificativo del tipo de archivo o programa, más el nombre del archivo, más la extensión en este caso propia del lenguaje de programación dinámico PHP.



modules

Formada por los distintos módulos y extensiones, o añadidos de las funciones básicas del sitio web. Por ejemplo un módulo para la traducción del sitio web, módulo gestión de archivos vía web-ftp, módulo de gestión de plantillas CSS para el sitio web, etc. Los archivos contenidos deberían tener la forma "mod.nombre.php", es decir, prefijo identificativo del tipo de archivo o programa, más el nombre del archivo, más la extensión en este caso propia del lenguaje de programación dinámico PHP.



install

En los sitios web más avanzados como los CMS suele ser necesario efectuar un proceso de instalación y configuración previa, antes de su uso. Los archivos que





		suelen guardarse en la carpeta "install".
	index.php	Página principal e inicial del sitio web a partir de la cual se accede a todos los contenidos. El único cambio que se produce con respecto a la estructura de la web estática es el tipo de archivo, la extensión y el lenguaje de programación utilizado.
	inc.head.php	Es muy frecuente que la página principal o inicial del sitio web esté compuesta por distintos fragmentos. La capacidad modular de PHP unido al potencial de las capas CSS para estructurar la información, hacen que el encabezamiento de la página se divida, generando el archivo "inc.head.php". El prefijo <i>inc</i> , indica que se trata de un "include" o añadido que se ensambla con la página principal. Seguido de un punto se indica el componente <i>head</i> y finalmente la extensión del archivo.
	inc.menu.php	Al igual que en el caso de la cabecera el menú también se modula para formar parte de un archivo más del ensamblaje final de la página principal.
	inc.foot.php	El pie de página también mantiene el mismo modelo de parcelación visto en el encabezado y el menú. Con todos estos componentes se configura no solamente la página inicial, sino cualquier otra página del sitio web manteniendo en todo momento la estética, organización, contextualidad y base estructural.

Figura 5. Muestra de estructura básica de un sitio web dinámico

12. Maquetación de textos con CSS

La correcta maquetación de textos, imágenes y listados ayuda a crear una presentación de los contenidos, limpia, clara y precisa. Por ello se recomienda tener control sobre sus principales propiedades, los márgenes, justificados, sangrados, alineaciones, proporciones, etc. Las hojas de estilo en cascada, también denominadas CSS, ayudan en este propósito y por ello se han reseñado las propiedades más importantes a tener en cuenta en el proceso de maquetación de textos.

Las propiedades del texto

Una de las claves para la correcta visualización de la información del sitio web es la definición de los atributos del texto de forma tal que sean perfectamente visibles y legibles en el sitio web. Se recomienda utilizar los atributos relativos al tipo de fuente (utilizar fuentes normalizadas con valores *arial*, *sans serif*, *serif*, *tahoma*, *verdana*, *times new roman*), tamaño de la fuente (utilizar medidas en *px*, *pt* o *em*), grosor de la fuente (utilizar valores *lighter* para letra normal o *bold* para letra negrita destacada), tipo de decoración (utilizar valores de subrayado o tachado) y color de la fuente (Utilizar código hexadecimal de color), véase tabla 13.

<pre>.texto {font-family: arial, sans serif; font-size: 12px; font-weight: lighter; font-style: normal; text-decoration: none; color: #000000;}</pre>

<p>W3Schools. 2015. CSS Text. [http://www.w3schools.com/css/css_text.asp]</p>

<p>W3Schools. 2015. CSS Font. [http://www.w3schools.com/css/css_font.asp]</p>

Tabla 13. Propiedades del texto

Encapsular los textos en capas

Una regla de oro para el control de los textos y contenidos del sitio web es su encapsulado dentro de una capa <div>. Si se desea repercutir un estilo determinado en un texto, se necesitará encapsularlo previamente, véase la tabla 14. Este método es aplicable a cualquier elemento textual, imagen, lista e incluso tablas, de cara a lograr una normalización en la representación de los contenidos.

```
<div class='texto'>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam auctor  
consectetur nisl, nec viverra sapien tincidunt ornare. Suspendisse potenti. In at interdum  
ligula. Vivamus ligula quam, varius eu placerat eu, porta eu mi. Nam nec felis orci, ac  
consectetur diam. Praesent quis odio in erat fermentum euismod at vitae odio. Nunc rhoncus  
dolor ac erat laoreet gravida.</div>
```

Tabla 14. Encapsulado de un texto en una capa

Las propiedades de las capas

A su vez las capas pueden tener sus propios estilos para la representación de los contenidos. Se recomienda determinar siempre que sea posible el ancho, el alto, los márgenes externos e internos, el color de fondo y el borde, véase “*capa1*” de la tabla 3. Además un determinado tipo de capa, puede incluir también las propiedades del texto, véase el estilo de “*capa2*” de la tabla 15.

```
.capa1 {width: 90%; height: auto; margin: 10px; padding: 10px; background-color: #f1f1f1;  
border: solid 2px #e2e2e2;}  
  
.capa2 {width: 50%; height: auto; margin: 10px; padding: 10px; background-color: #e2e2e2;  
border: solid 2px #d4d4d4; font-family: arial, sans serif; font-size: 14px; font-weight:  
lighter; font-style: normal; text-decoration: none; color: #000000;}
```

W3Schools. 2015. CSS Background. [http://www.w3schools.com/css/css_background.asp]

W3Schools. 2015. CSS Border. [http://www.w3schools.com/css/css_border.asp]

Tabla 15. Ejemplo de propiedades de las capas

Márgenes

Todos los textos del sitio web deberán constar de márgenes internos y externos, ello se establece con los atributos CSS “*margin*” (margen externo) y “*padding*” (margen interno). Véase tabla 16. El espaciado mínimo recomendado en cada margen deberá ser de “*10px*” como mínimo.

```
<div style='margin: 20px; padding: 30px;'>texto</div>
```

W3Schools. 2015. CSS Margin. [http://www.w3schools.com/css/css_margin.asp]

W3Schools. 2015. CSS Padding. [http://www.w3schools.com/css/css_padding.asp]

Tabla 16. Márgenes internos y externos

Justificados

La alineación de los textos, siempre que sea posible, deberá de ser justificada a ambos lados del margen, por lo que el atributo “*text-align*” tomará el valor “*justify*”. Véanse las distintas opciones de alineación en la tabla 17.

<pre><div style='text-align: left;'>texto</div> <div style='text-align: right;'>texto</div> <div style='text-align: center;'>texto</div> <div style='text-align: justify;'>texto</div></pre>
W3Schools. 2015. CSS Text. [http://www.w3schools.com/css/css_text.asp]

Tabla 17. Alineación de textos a derecha, izquierda, centro o justificado

Sangrados

Opcionalmente, se recomienda el sangrado del comienzo de cada apartado, capítulo o sección de la información a fin de determinar que la temática abordada resulta distinta a la del párrafo anterior. Ello se consigue mediante el atributo “*text-indent*”. Un valor habitual se sitúa a partir de “*30px*”, véase tabla 18.

<pre><div style='text-indent: 30px;'>texto sangrado</div></pre>
W3Schools. 2015. CSS Text. [http://www.w3schools.com/css/css_text.asp]

Tabla 18. Sangrado de textos

Control de dimensiones

A veces el texto que es contenido dentro de una capa, está embebido en otra a su vez. En tales casos es necesario determinar que anchura, altura y en definitiva dimensiones tendrá. Ello se consigue determinando el ancho y alto de la capa contenedora del texto, véase tabla 19.

<pre><div style='width: 80%;'>texto</div> <div style='width: 200px;'>texto</div></pre>
W3Schools. 2015. CSS Dimension. [http://www.w3schools.com/css/css_dimension.asp]

Tabla 19. Dimensiones de las capas

Imágenes y texto

En muchos casos el texto acompaña imágenes e ilustraciones. En esa situación se necesita establecer un formato flotante si se quieren alinear los textos a las imágenes, ya sea a izquierda o a derecha, véase tabla 20. Se recomienda siempre establecer un margen externo para evitar que el texto quede pegado a la imagen, siendo dicho margen superior a “10px”.

```
<div style='width: 50%;'>  
<img src='#' style='float: left; margin: 20px;'/> texto  
</div>
```

Tabla 20. Ejemplo de integración de imágenes y texto dentro de la misma capa

Pero por lo general el método más claro y efectivo de presentar la información es mostrando las imágenes en espacios independientes al texto, ocupando todo el ancho de la página o estructura div que se haya predeterminado. En estos casos se hace necesario especificar que no puede haber capas flotantes ni a derecha, ni a izquierda, para ello se utiliza el atributo “clear” con el valor “both”, véase tabla 21.

```
<div><img src='#' style='clear: both; margin: 10px;'>pie de foto</div>  
<div style='clear: both;'>texto</div>
```

W3Schools. 2015. CSS Float. [http://www.w3schools.com/css/css_float.asp]

Tabla 21. Ejemplo de imágenes y textos en capas independientes

Listas

Las listas deberán guardar algunos de los estilos presentados en la tabla 22. No se recomienda el uso de imágenes o viñetas para cada elemento de la lista. Además se recomienda encapsular las listas en una capa “<div>” con márgenes internos y externos.

```
ul.circulo {list-style-type: circle;}  
ul.cuadrado {list-style-type: square;}  
ol.romanos {list-style-type: upper-roman;}  
ol.alfabetico {list-style-type: lower-alpha;}
```

```
<div style='margin: 10px; padding: 10px;'>  
  <ul class='circulo'>  
    <li>Opción desordenada 1</li>  
    <li>Opción desordenada 2</li>  
    <li>Opción desordenada 3</li>  
  </ul>
```

<pre> </div> <div style='margin: 10px; padding: 10px;'> <ol class='romanos'> Opción ordenada 1 Opción ordenada 2 Opción ordenada 3 </div> </pre>
<p>W3Schools. 2015. CSS List. [http://www.w3schools.com/css/css_list.asp]</p>

Tabla 22. Aplicación de estilos en listas ordenadas y desordenadas

Tablas

Para toda tabla que se edite, se recomienda reseñar los atributos de ancho máximo, borde, tipo de borde, ancho de borde, color de borde y separación colapsada. Para la fila de encabezamiento de las columnas de la tabla “<th>” se recomienda aplicar un estilo diferenciador con respecto a las filas que contendrán los datos y textos “<td>”, véase tabla 23.

<pre> .tablas {width: 90%; border: solid 2px #cccccc; border-collapse: collapse;} .filas1 {padding:15px;} .filas2 {padding:10px;} </pre>
<pre> <table class='tablas'> <tr><th class='filas1'>título de columna</th></tr> <tr><td class='filas2'>fila con datos</td></tr> <tr><td class='filas2'>fila con datos</td></tr> <tr><td class='filas2'>fila con datos</td></tr> <tr><td class='filas2'>fila con datos</td></tr> </table> </pre>
<p>W3Schools. 2015. CSS Table. [http://www.w3schools.com/css/css_table.asp]</p>

13. Diseño de estructuras web: iframe vs include

Desde un punto de vista estructural, los sitios web se desarrollan a partir de capas que dividen las distintas áreas que las componen, a saber: encabezamiento, menú general, menú local, contenidos y navegación contextual y pie de página, véase figura 6. Recuérdese también el apartado de diseño de estructuras web con capas y CSS layouts. En el presente artículo se abordará la forma de interacción, navegación y enlace de los contenidos del sitio web dentro de la estructura.

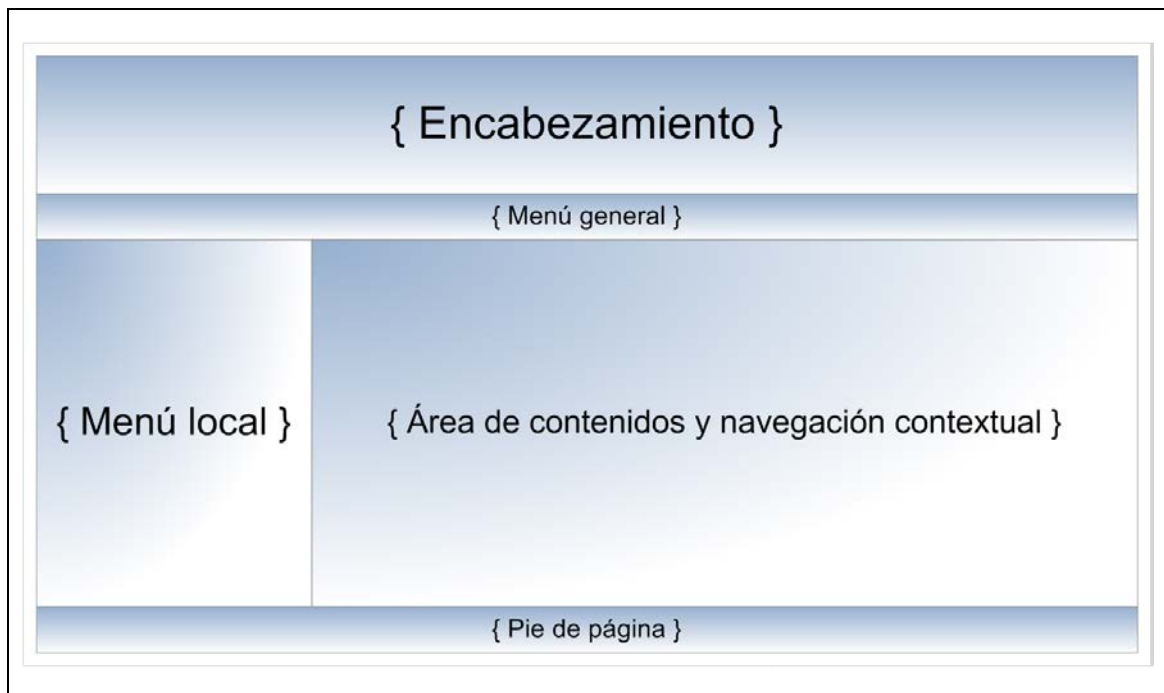


Figura 6. Áreas clave de la estructura de un sitio web

Método iframe

El método iframe (Incorporated Frame = Marco Incorporado) consiste en embeber un marco o ventana dentro del área de contenidos del sitio web, ocupando el 100% del ancho y el 100% del alto disponible en el espacio. De esta forma todo contenido del sitio web será cargado en dicho espacio. Una de sus ventajas es que el contenido cargado en un iframe es independiente del resto, lo que ofrece una gran flexibilidad para cargar contenidos de cualquier tipo. También su método de enlazamiento resulta muy sencillo y efectivo. Todos los “*iframe*” disponen de un atributo “*name*” con el que es posible identificarlos correctamente. Estos nombres pueden ser utilizados como los

destinos de carga de las páginas de contenido enlazadas desde un menú general o local, véase tabla 23 y 24.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>
<html>
<head>
<title>Ejemplo de estructura iframe</title>
<meta http-equiv='content-type' content='text/html; charset=UTF-8'/>

<style>
.capa1 {width: 100%; height: 100px; clear: both; background-color: #e0e0e0;}
.capa2 {width: 30%; height: 600px; float: left; background-color: #dedede;}
.capa3 {width: 70%; height: 600px; float: left; background-color: #ffffff;}
.capa4 {width: 100%; height: 50px; clear: both; background-color: #e0e0e0;}
.contenido {padding: 10px;}

ol.romanos {list-style-type: upper-roman;}
li {margin: 10px;}

a:link { color: #224a6a; font-size: 0.9em; font-family: arial; }
a:visited { color: #236b6a; font-size: 0.9em; font-family: arial; }
a:hover { color: #4497da; font-size: 0.9em; font-family: arial; }
a:active { color: #44c5da; font-size: 0.9em; font-family: arial; }

</style>

</head>
<body>

<div class='capa1'>
<div class='contenido'>texto1</div>
</div>

<div class='capa2'>
<div class='contenido'>
<div>
<ol class='romanos'>
<li><a href='http://www.bne.es/' target='visor'>Biblioteca Nacional de España</a></li>
<li><a href='http://www.loc.gov/' target='visor'>The Library of Congress</a></li>
<li><a href='http://www.bl.uk/' target='visor'>The British Library</a></li>
<li><a href='http://www.bnf.fr/' target='visor'>Bibliothèque nationale</a></li>
<li><a href='http://www.d-nb.de/' target='visor'>Deutschen Nationalbibliothek</a></li>
<li><a href='http://www.kb.nl/' target='visor'>Koninklijke Bibliotheek</a></li>
</ol>
</div>
</div>
</div>

<div class='capa3'>
<iframe src='#' name='visor' style='width: 100%; height: 100%; border: 0;'/>
</div>

<div class='capa4'>
<div class='contenido'>texto4</div>
</div>

</body>
</html>

```

Ejemplo de estructura iframe. mblazquez.es. Disponible en:
<http://www.mblazquez.es/blog-ccd-oc-arquitectura-informacion/documentos/structure-iframe.html>

Tabla 23. Sitio web con método iframe para cargar páginas y contenidos

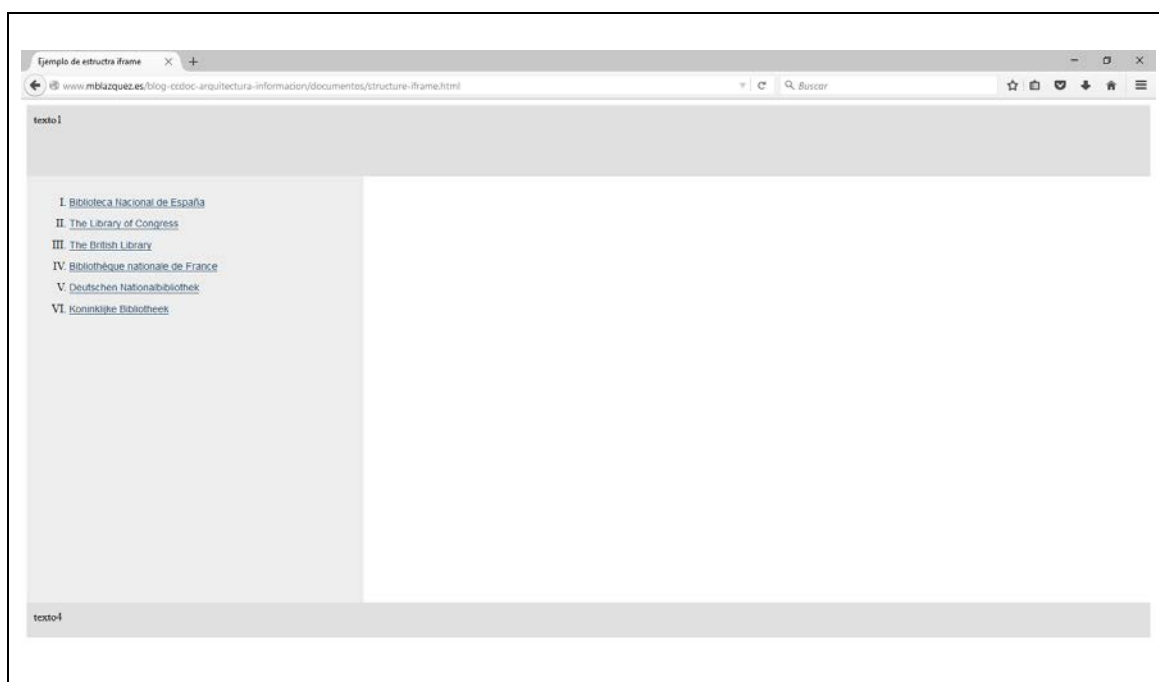


Tabla 24. Ejecución en el navegador del código fuente de la tabla 23

No obstante, el método `iframe`, pese a sus grandes ventajas como la simplicidad y sencillez que aporta al conjunto final de la superestructura de la web, tiene inconvenientes. Por ejemplo, aun siendo un elemento aceptado por la comunidad de desarrolladores y ampliamente utilizado para la elaboración de aplicaciones AJAX (Asynchronous JavaScript and XML), está considerado como un elemento poco accesible, ya que al generar un marco para cargar contenidos hace que también sean independientes del resto de la estructura, no estando integrados en un único código fuente. Esto hace que sea difícil la lectura para cualquier sistema especial de interpretación para personas con discapacidad. Otro inconveniente, esta vez de diseño, es relativo a la altura de la página. Normalmente la altura de una página de contenido suele ser variable y resulta recomendable su visualización completa sin scroll. Con `iframe` es necesario establecer la altura de forma absoluta, haciendo que los contenidos visualizados dentro de su ventana activen en muchas ocasiones su correspondiente scroll vertical. Estas cuestiones han llevado a muchos desarrolladores a elaborar páginas web mucho más sencillas, también de tipo estático o páginas web mucho más funcionales, modulares y completas de tipo dinámico mediante el método `include`.

Método include

Consiste en el empleo de una función del lenguaje PHP denominada include que es utilizada para incrustar páginas web en formato HTML o scripts en lenguaje PHP. Esta capacidad de incluir el código fuente de distintas páginas web y combinarlo en una, permite una estructuración perfecta del sitio web, sin necesidad de emplear iframe. La forma más común de componer la estructura del sitio web es creando un archivo matriz en PHP que contenga un include por cada área de la estructura, por ejemplo un include con el archivo que contiene el código del encabezado, un include para el archivo que contenga el menú general y otro para el pie de página. Véase la tabla 25 con el ejemplo de uso de include y de los distintos archivos auxiliares con el código del sitio web.

```
<?php
// archivo matriz: structure-index.php
// =====

include("structure-inc.head.php"); // Carga código de encabezamiento
include("structure-inc.menu.php"); // Carga menú general
echo "<div>Emplazamiento de los contenidos</div>";
include("structure-inc.foot.php"); // Carga el pie de página

?>
```

```
<?php
// archivo auxiliar: structure-inc.head.php
// =====

echo "

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>
<html>
<head>
<title>Ejemplo de estructura iframe</title>
<meta http-equiv='content-type' content='text/html; charset=UTF-8'/>

    <style>
    .capa1 {width: 100%; height: 100px; clear: both; background-color: #e0e0e0;}
    .capa2 {width: 100%; height: 50px; clear: both; background-color: #ededed;}
    .capa3 {width: 100%; height: 50px; clear: both; background-color: #e0e0e0;}
    .contenido {padding: 10px;}
    </style>

</head>
<body>

<div class='capa1'>
<div class='contenido'>encabezamiento</div>
</div>

";
?>
```

```
<?php
// archivo auxiliar: structure-inc.menu.php
// =====
```

<pre> echo " <div class='capa2'> <div class='contenido'>menú general</div> </div> "; ?> </pre>
<pre> <?php // archivo auxiliar: structure-inc.foot.php // ===== echo " <div class='capa3'> <div class='contenido'>pie de página</div> </div> </body> </html> "; ?> </pre>

Tabla 25. Ejemplo de aplicación del método include para componer sitios web

Se observará en el ejemplo cómo se combina la programación PHP y HTML para dar forma al sitio web. De hecho programar un sitio web en lenguaje PHP implica conocimientos profundos de HTML. Esto significa que la forma de representar los contenidos siempre es la misma, pero la manera en que se conjuga la información varía, adquiriendo una funcional propiedad de modularidad. Se pueden crear tantas páginas de contenido como se deseen, con la estructura principal del sitio web, siempre que se mantengan las instrucciones include con la referencia a los archivos auxiliares que portan la estructura. Véase el resultado de aplicar el código de la tabla 25 en la siguiente tabla 26.


<p>Ejemplo de estructura iframe. Disponible en: http://www.mblazquez.es/blog-ccdod-arquitectura-informacion/documentos/structure-index.php</p>
<p>Descargar código fuente. Disponible en: http://www.mblazquez.es/blog-ccdod-arquitectura-informacion/documentos/structure-includes.zip</p>

Tabla 26. Ejecución en el navegador del código de la tabla 25

Vistas las ventajas, también se observan algunos inconvenientes. Por ejemplo PHP es un lenguaje de programación del lado del servidor. Esto significa que para poder ejecutarse este tipo de páginas se requiere de un servidor web con unas especificaciones muy concretas:

- Apache HTTP Server (Servidor dónde se aloja el sitio web programado en PHP)
- Compilador PHP (Intérprete del lenguaje PHP)
- MySQL (Recomendable, pero no necesario para ejecutar PHP)

Con HTML no era necesario este despliegue de medios, pero también son mayores las limitaciones. Por suerte, existen muchos paquetes y distribuciones de instalación de los componentes referidos denominados “*de tipo AMP*” como por ejemplo AMPdoc 2.0, disponible en: <http://mblazquez.es/ampdoc-2-0/>

14. Adaptación dinámica de las páginas de contenidos

Hasta el momento se ha trabajado con páginas de contenido estáticas basadas en HTML. Pero como se pretende enseñar, estas páginas pueden ser adaptadas para su aprovechamiento en la estructura del sitio web dinámico en PHP. Tomando un contenido estático, no resulta difícil averiguar que cada página de contenido es una página matriz en la que se cargan mediante include el encabezado, los menús y el pie de página.

Ejemplo de adaptación

En la siguiente tabla, se muestra un ejemplo de cómo debe ser adaptada una página de contenidos. En primer lugar se parte de una página matriz tipo, que permita cargar el encabezado *"inc.head.php"*, el menú general *"inc.menu.php"* y el pie de página *"inc.foot.php"*. A partir de este modelo, el código fuente correspondiente a los textos e informaciones de la página, se introducirá entre el menú y el pie de página, ya que es el espacio o área de la estructura destinada al contenido. De esta forma, se logra componer todas las piezas correctamente.

```
<?php
// Página de contenidos = Página matriz
// =====

include("inc.head.php");
include("inc.menu.php");

echo "
<div style='padding: 20px;'"

<h1>Titular</h1>

<h2>Resumen</h2>
<div class='textos'>texto del resumen</div>

<h2>Autor</h2>
<div class='textos'>Apellidos, Nombre</div>
<div class='textos'>Institución</div>
<div class='textos'>Datos de filiación</div>
<div class='textos'>Correo electrónico</div>

<h2>Versión</h2>
<div class='textos'>Última versión</div>
<div class='textos'>Versiones anteriores</div>

<h2>Tabla de contenidos</h2>
<div class='textos'>
```

```

<ol>
  <li><a href='#p1' title='titulo'>Apartado 1</a></li>
  <li><a href='#p2' title='titulo'>Apartado 2</a></li>
  <li><a href='#p3' title='titulo'>Apartado 3</a></li>
  <li><a href='#p4' title='titulo'>Apartado 4</a></li>
</ol>
</div>

<h2><a name='p1'>Apartado 1</a></h2>
<div class='textos'>Texto del apartado 1</div>

<h2><a name='p2'>Apartado 2</a></h2>
<div class='textos'>Texto del apartado 2</div>

<h2><a name='p3'>Apartado 3</a></h2>
<div class='textos'>Texto del apartado 3</div>

<h2><a name='p4'>Apartado 4</a></h2>
<div class='textos'>Texto del apartado 4</div>

</div>
";
include("inc.foot.php");
?>

<?php
// inc.head.php
// =====

echo "
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>
<html>
<head>
<title>Ejemplo de estructura iframe</title>
<meta http-equiv='content-type' content='text/html; charset=UTF-8'/>

<style>
.capa1 {width: 100%; height: 100px; clear: both; background-color: #e0e0e0;}
.capa2 {width: 100%; height: 50px; clear: both; background-color: #ededed;}
.capa3 {width: 100%; height: 50px; clear: both; background-color: #e0e0e0;}
.contenido {padding: 10px;}

h1 {font-family: verdana; font-size: 18px;}
h2 {font-family: verdana; font-size: 14px;}
th {padding: 5px; font-family: verdana;}
td {padding: 5px; font-family: verdana;}

.textos {font-family: verdana; font-size: 12px; text-align: justify; color: #000000; line-
height: 25px;}

.imagenes {margin: 10px; padding: 10px; text-align: center; background-color: #eeeeee; border:
solid 2px #cccccc; border-radius: 15px;}

.tablas {margin: 0 auto; background-color: #f2f2f2; border: outset 2px #c0c0c0;}

a:link {font-family: verdana; font-size: 12px;}
a:visited {font-family: verdana; font-size: 12px;}
a:hover {font-family: verdana; font-size: 12px;}
a:active {font-family: verdana; font-size: 12px;}
</style>

</head>
<body>

<div class='capa1'>
<div class='contenido'>encabezamiento</div>

```

<pre></div> "; ?></pre>
<pre><?php // inc.menu.php // ===== echo " <div class='capa2'> <div class='contenido'>menú general</div> </div> "; ?></pre>
<pre><?php // inc.foot.php // ===== echo " <div class='capa3'> <div class='contenido'>pie de página</div> </div> </body> </html> "; ?></pre>
<p>Descargar código fuente. Disponible en: http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/practica5-ejemplo.zip</p>

Tabla 27. Ejemplo de adaptación dinámica de una página con el método include

15. Arquitectura de formularios y su adaptación dinámica

Los formularios constituyen un elemento esencial en todo el aparato de comunicaciones y edición del sitio web. Es por ello que requieren un tratamiento especial y una revisión en su modo de construcción. Los formularios son empleados como recipiente de recogida de datos. Su modo empleo más común es el envío de mensajes y correos como método de contacto.

Pero también son utilizados para otros cometidos, por ejemplo, como sistema de selección y filtrado de opciones en un buscador, como campos de edición para un registro, para representar datos de una edición, para enviar valores y contenidos entre distintas páginas web y finalmente el uso dinámico de la información que el usuario introduce en el sitio web. Dicho de otra forma, supone uno de los nexos que conectan el sistema de información al usuario.

Formularios estáticos

Se considera formulario estático aquel que no emplea funciones de PHP para la recogida de información enviada por sí mismo o a otra página o programa de PHP (Léase cualquier programa elaborado con un lenguaje de programación dinámico orientado a objetos).

Ello significa que un formulario estático es únicamente una estructura de etiquetas HTML con una disposición básica en cuanto a su modo de funcionamiento (es decir, un modo de funcionamiento sencillo, unidireccional a un determinado destino que se identifica claramente con un correo electrónico), por ejemplo véase tabla 28.

```

<form name='contactar' id='contactar' action='mailto:mblazquez.ccdoc@gmail.com' method='post'>
<input type='hidden' name='id' value=''/> identificador
<input type='text' size='40' name='titulo' value=''/> título
<input type='password' size='40' name='pass' value=''/> contraseña

<input type='radio' name='centro' value='archivo'> Archivo
<input type='radio' name='centro' value='biblioteca'> Biblioteca
<input type='radio' name='centro' value='museo'> Museo

<input type='checkbox' name='user' value='investigador'> Investigador
<input type='checkbox' name='user' value='profesor'> Profesor

<textarea cols='60' rows='4' name='comentarios'></textarea>

<select name='tipologia'>
  <option value='monografia'>Monografía</option>
  <option value='diccionario'>Diccionario</option>
  <option value='articulo'>Artículo</option>
</select>

<select name='tematica'>
  <optgroup label='Ciencias y técnicas auxiliares'>
    <option value='bibliofilia'>Bibliofilia</option>
    <option value='bibliografia'>Bibliografía</option>
    <option value='encuadernacion'>Encuadernación</option>
  </optgroup>
  <optgroup label='Museología'>
    <option value='museografia'>Museografía</option>
    <option value='patrimonio'>Patrimonio</option>
  </optgroup>
</select>

<input type='button' onClick="history.go(-1)" value='atrás'/>
<input type='button' onClick="history.go(1)" value='adelante'/>
<input type='button' onClick="window.location='http://'" value='cargar página'/>
<input type='button' onClick="window.location.reload()" value='recargar página actual'/>

<input type='submit' name='go' value='Enviar datos'/>

</form>

```

W3Schools. 2015. HTML Forms. [http://www.w3schools.com/html/html_forms.asp]

Tabla 28. Principales componentes del formulario

Elemento	Descripción
<code><input type='text'.../></code>	Elemento principal de un formulario. La caja de texto es empleada para almacenar 1 línea de texto o cadena de caracteres.
<code><input type='hidden'.../></code>	Caja de texto oculta en el formulario, no visible para el usuario. Ampliamente utilizada para almacenar valores especiales, identificadores de registros, o cualquier información necesaria para la ejecución de una rutina en PHP. Es muy frecuente utilizar un campo oculto para alojar el valor del identificar de un registro de un documento, o un código alfanumérico que debe ser utilizado posteriormente para efectuar una consulta o un

	valor que debe ser almacenado con fines estadísticos, de contabilización, por ejemplo.
<code><input type='password'.../></code>	Campo de texto confidencial destinado a la introducción de contraseñas, evita que se visualicen en pantalla y bloquea la copia del valor de la contraseña que se escribe. Utilizado frecuentemente en el desarrollo de aplicaciones de acceso a páginas web.
<code><input type='radio'.../></code>	Campo de selección por círculo, permite establecer distintas opciones entre las que se debe seleccionar una de ellas.
<code><input type='checkbox'.../></code>	Selección de una o varias opciones mediante cajas de verificación con ticks.
<code><textarea...></textarea></code>	Campo para el área de texto. Cuando se necesita reseñar textos más largos de una línea, se utiliza este elemento.
<code><select...> <option...></option> </select></code>	Campo selector que permite escoger una opción entre las presentadas en la lista.
<code><select...> <optgroup...> <option...></option> <optgroup> </select></code>	Campo selector, con opciones agrupadas.
<code><input type='button'.../></code>	Campo de botón, utilizado para añadir eventos JavaScript, funcionando como método desencadenante de acciones. Por ejemplo, cargar una página web al hacer clic, actualizar la página actual al hacer clic, etc.
<code><input type='submit'.../></code>	Botón de envío de formularios.

Tabla 29. Descripción de componentes del formulario

El método de envío de datos

La forma o protocolo con el que se envía la información es fundamental dependiendo del uso que se hace de la información contenida en los mensajes. Recuérdese que muchos formularios, actúan como interfaz entre el usuario y el sistema, para que éste actúe de una determinada forma o simplemente para que publique una información o sea enviado a un determinado destino. Para ello los mensajes escritos desde formularios pueden ser enviados mediante método GET, reconocible e identificable mediante una

cabecera pública o bien por el método POST, cuya información queda asegurada hasta su recepción en el servidor de destino.

Método GET

Permite efectuar una solicitud al servidor de tal forma que ésta pueda ser identificada mediante URI. Los valores y datos del mensaje se envían desde la cabecera y no dentro del cuerpo del mismo, por ello quedan expuestos y se considera un método de comunicación menos seguro.

Método POST

Permite efectuar una solicitud al servidor de forma segura sin posibilidad de identificar el recurso enviado por URI. Esto se debe a que los datos y variables de la petición se encuentran en el cuerpo del mensaje y no en la cabecera. Por ejemplo, esto significa que los datos enviados en un proceso de comunicación como en un grupo de noticias, una lista de correos, un comentario, un correo electrónico o la petición de publicación de información no pueden ser identificados a excepción de la página del servidor al que se destine la solicitud.

El formulario dinámico

La información enviada a través de formularios ya fuere por método POST o GET, puede ser recuperada y gestionada mediante variables globales propias de los lenguajes de programación dinámicos, por ejemplo PHP. De tal forma que si se dispone de un formulario como el presentado en la tabla 30 y un archivo de recepción de los datos del formulario, como el dispuesto en la tabla 31, sería posible enviar mensajes e imprimirlos en pantalla.

```
<!-- archivo: enviar.html -->
<!-- ===== -->
<form action='recibir.php' method='post'>
  <input type='text' name='title' />
  <textarea cols='60' rows='3' name='description'></textarea>
  <input type='submit' name='send' value='Publicar' />
</form>
```

Tabla 30. Formulario para el envío de datos con método POST

```
<?php
// archivo: recibir.php
// =====
echo "<div>";
echo "<div>$_POST[title]</div>";
echo "<div>$_POST[description]</div>";
echo "</div>";
?>
```

Tabla 31. Archivo de recepción que presenta en pantalla la información del mensaje

De esta forma es posible utilizar la información enviada para que sea almacenada, mostrada en pantalla, enviar parámetros de ejecución para un programa o simplemente para enviar un correo electrónico. En PHP se utilizan las variables globales “\$_GET[nombre]” y “\$_POST[nombre]” en alusión al método por el que fue enviada la información del formulario y el nombre del parámetro que fue enviado desde una caja de texto, área de texto, selector o cualquier otro elemento del formulario. En PHP una variable es un recipiente utilizado para almacenar información, datos, textos o valores de tipo numérico, textual o alfanumérico. De esta forma con independencia del tipo de formulario empleado la información enviada es empaquetada y guardada en estas variables de tal forma que puedan ser recuperadas y posteriormente utilizadas. Se denominan variables globales, porque pueden ser utilizadas en cualquier punto del programa PHP que se edite para recibir la información del formulario. No obstante también tienen caducidad si no son recibidas por el destinatario, perdiéndose la información. Por otra parte, también es práctica común utilizar una sola página para el envío, recepción y proceso de la información, véase tabla 32.

```
<?php
// archivo: enviar-recibir.php
// =====
echo "<form action='enviar-recibir.php' method='get'>
  <input type='text' name='title' />
  <textarea cols='60' rows='3' name='description'></textarea>
  <input type='submit' name='send' value='Publicar' />
</form>
";

if($_GET[send]){  echo "<div>";
  echo "<div>$_GET[title]</div>";
  echo "<div>$_GET[description]</div>";
  echo "</div>";
}
?>
```

Tabla 32. Archivo de envío y recepción de datos

Obsérvese que en la tabla 32, a diferencia de los casos anteriores, se añade una estructura de control condicional para desencadenar la impresión en pantalla del

mensaje solamente si este fue enviado. En este código la condición “*if(condición){instrucción}*” establece entre paréntesis que la variable “*send*” debió de ser enviada, es decir, el usuario debió de hacer clic en el botón publicar. Si esa condición se cumple, se ejecuta todo el proceso que ha sido alojado entre la apertura y cierre de llaves. En este caso es la impresión del campo título y campo descripción del formulario.

A pesar de lo expuesto, existen otras formas de enviar información, no únicamente mediante formularios. Se trata del caso del método GET por medio de URL. Tal como se explicaba anteriormente, el protocolo GET implica una identificación mediante URI (Universal/Uniform Resource Identifier) es decir, que la información es enviada a través de una URL y es visible por aparecer en la cabecera de las peticiones que se envían mediante HTTP. De hecho cuando se aprecian direcciones como las de la tabla 33, se aludirá al método GET, por medio de URL.

```
http://www.dominio.org/publicaciones.php?id=10&url=biblioteca&title=fuentes-  
bibliotecarias&text=Esto%20es%20una%20prueba  
  
<?php  
  echo "$_GET[id]";  
  echo "$_GET[url]";  
  echo "$_GET[title]";  
  echo "$_GET[text]";  
?>
```

Tabla 33. Archivo de envío y recepción de datos

La información puede ser recuperada igualmente, con la salvedad de que ésta debe ser emplazada con una sintaxis determinada. Como se puede observar la URL separa el nombre y extensión del archivo de los valores o datos que serán transmitidos por medio de un signo de interrogación “?”. Seguidamente se reseñan en color azul el nombre de la variable que contendrá un determinado dato, separado por el signo igual “*variable=dato*”. Cuando existen más de una variable a transmitir, se emplea el signo ampersand “&” para distinguir la primera variable de las subsiguientes “*variable1=dato1&variable2=dato2*”.

16. Navegación dinámica vs navegación estática

Hasta el momento se han abordado aspectos fundamentales de la construcción de un sitio web, desde su concepción, organización, estructura formal y técnica, su programación óptima, aspectos de estilo y formato, pero no se han atendido otros que resultan cruciales, como es la forma de navegación y enlazamiento del sitio web.

Tipos de enlaces

Antes de explicar los métodos de navegación estática y dinámica, se deben observar la tipología de enlaces que con frecuencia se utilizan en toda la web, véase tabla 34. Los más conocidos y utilizados son los enlaces de tipo absoluto que o bien pueden apuntar y enlazar a una página web dinámica (programada en PHP, ASP, Ruby, Python o cualquier otro lenguaje de programación del lado del servidor) o a una página web estática (normalmente programada en HTML). Se denomina enlace absoluto cuando se identifica en la construcción de la URL el protocolo de internet “*http://www.*” añadiendo el nombre del dominio, subdominio, extensión “*dominio.org*” y agregando finalmente la ruta de carpetas o repositorios hasta localizar el archivo o documento a enlazar “*document001.php*”.

Otro tipo de enlaces básico son los denominados enlaces relativos. Éstos son de uso común en la práctica totalidad de los sitios web. Se utilizan para enlazar contenidos sin necesidad de reseñar la ruta completa o absoluta hasta el recurso o documento objetivo. Esto implica una reducción importante en la longitud de la URL y una simplificación en su construcción. Un enlace relativo puede ser utilizado para enlazar un documento que se encuentra en el mismo origen o nivel del que parte el link, por ejemplo (Un enlace como este `documento2` dentro del código de una página denominada `document001.html` implica que tanto el `documento1` como el `2` están dentro del mismo nivel de carpetas, con independencia de su posición en la estructura de repositorios del sitio web). Cuando el recurso que se pretende enlazar se encuentra en un nivel jerárquico de carpetas inferior al que se encuentra el origen del que parte el link se reseña la carpeta/s a las que son necesarias acceder hasta el recurso objetivo, creando una ruta más elaborada, por ejemplo “*documents/document001.html*”. Pero en el caso en el que el recurso a enlazar se encuentre en un nivel de carpetas superior, o en otro nivel distinto, se hace necesario especificar mediante la sintaxis (*../*)

que la ruta de acceso ha de subir un nivel de carpetas o tantos como fueren necesarios (../.../ 3 en este caso) para acceder al documento o carpetas en las que está alojado el documento, por ejemplo (../css/estilo.css).

Enlaces	Descripción
http://www.dominio.org/document001.php	Modelo-A; absoluto; dinámico; no-nivel. Enlace absoluto a una página web dinámica.
http://www.dominio.org/document001.html	Modelo-A; absoluto; estático; no nivel. Enlace absoluto a una página web estática.
documents/document001.html	Modelo-B; relativo; estático; N-superior. Enlace relativo desde el nivel raíz del sitio web a un documento dentro de una carpeta denominada documents.
../css/estilo.css	Modelo-B; relativo; estático; N-inferior. Enlace relativo desde un primer nivel de carpetas hasta la carpeta "css" que contiene el archivo "estilo.css"
http://www.dominio.org/document001.php?id=243&section=biblioteconomia	Modelo-C; absoluto; dinámico; no-nivel; variables GET. Enlace absoluto a una página web dinámica con variables enviadas por método GET.
document001.php?id=142&section=musicologia	Modelo-C; relativo; dinámico; N-igual; variables GET. Enlace relativo a una página web dinámica con variables enviadas por método GET.
index.php?page=documentacion-y-bibliotecas	Modelo-D; relativo; dinámico; N-igual; variables GET + SEO. Enlace a página web dinámica "index.php" con página de destino predefinida en variable "page" con nombre optimizado para SEO Search Engine Optimization.
index.php/documentacion-y-bibliotecas	Modelo-E; relativo; dinámico; N-igual; mod rewrite + SEO Enlace

<p>index.php/calendarios-de-transferencia</p>	<p>relativo a página web dinámica "index.php" con ruta dirigida por medio de función Mod rewrite htaccess, optimización de enlaces para SEO (Search Engine Optimization) en la configuración del servidor que aloja el sitio web.</p> <p>Modelo-F; relativo; dinámico; N-igual; parser URL + SEO. Enlace relativo a página web dinámica "index.php" similar al <i>modelo-E</i> pero cuya ruta no es dirigida por el servidor del alojamiento del sitio web, sino por un programa gestor de redireccionamientos integrado dentro del propio sitio web y programado en PHP o cualquier otro lenguaje del lado del servidor</p>
--	---

Tabla 34. Tipología de enlaces de la web

Pero como se ha visto en el artículo anterior relativo a formularios, se puede empezar a pensar en que uno de los tipos de enlaces más importantes es el que incorpora variables, que por lo general suele ser dinámico, ya que la página de destino, debe ser capaz de procesarlas, extraer sus datos, así "*document001.php?id=142§ion=musicología*" el archivo "*document001.php*" deberá ser capaz de entender que "*id=142*" corresponde a un identificador de un contenido sobre "*section=musicologia*".

Otro tipo de enlace derivado del "*modelo-C*" es el "*modelo-D*". Consiste en utilizar una única variable por método GET, que identifique el título completo o casi completo del artículo o documento objetivo. Si bien el enlace apunta a la página "*index.php*", ésta contendrá algún tipo de script o rutina programada capaz de interpretar el redireccionamiento que debe efectuar para cargar la página especificada por la variable "*page*". Esto significa que deberá identificar el documento por su título normalizado y además éste deberá ser único y distinto del resto de la colección de documentos o de la base de conocimiento, también deberá determinar la extensión del documento.

El enlace "*modelo-E*" corresponde a un método de redireccionamiento a páginas web que se resuelve en el lado del servidor mediante la activación de la función `mod_rewrite`

propia de los servidores HTTP de tipo Apache. Se lleva a cabo un proceso de reescritura de la URL de un recurso, documento o página web de acuerdo a un patrón establecido, por ejemplo la supresión de extensiones, el acortamiento o transformación del nombre del archivo por el del título del recurso, la adición de nuevos elementos en la ruta original (por ejemplo, fecha, nombre de un repositorio o sección), etc.

Finalmente el “*modelo-F*”, apenas se podría distinguir del “*modelo-E*”, si no fuera por el soporte de interpretación, que es distinto al explicado anteriormente. Si bien solamente aparece el título del documento o recurso objetivo, éste es interpretado por un programa denominado “*Gestor de enlazamiento y redirección para la navegación*”. Se trata de un ingenio que es capaz de analizar la estructura de un enlace y descomponerla en todos elementos. Una vez éstos son identificados se reconstruye la URL correspondiente al recurso objetivo.

17. Gestores de enlazamiento y redirección para la navegación

Un gestor de enlazamiento es un programa especializado en generar o reconstruir direcciones URL válidas a partir de variables, pistas, nombres, cadenas de texto, identificadores o enlaces orientados al SEO. De hecho no siempre el enlace visible, amigable y fácilmente comprensible para la lectura del usuario puede ser interpretado, por el sistema de información. Esto implica no sólo la capacidad de construir la URL sino de establecer una estructura de decisiones condicionada por el origen desde la página en la que se efectúa el enlace “*citante*” y efectuar la redirección a la página de destino especificada “*citada*”.

Todos estos aspectos deben ser resueltos para crear una experiencia de navegación adecuada en sitios web dinámicos. Un ejemplo básico de gestor de enlazamiento y redirección es el que se muestra en la tabla 1, utilizado para los modelos de enlace C/D, que contienen variables GET con los valores del documento enlazado. En tal caso el programa es capaz de determinar con cada variable y valor la página o documento que debe cargar o a la que debe redireccionar.

Aunque en el ejemplo de la tabla 35, se da por sentado que los contenidos se encuentran en forma de documento o página web generada en archivo con extensión “.*php*” o “.*html*”, también cabe la posibilidad (muy frecuente) de que la página dinámica se reedita mediante un visor de contenidos en PHP. Esto implica el uso de bases de datos MySQL en las que se almacena la información que se muestra al usuario. Por ejemplo una variable GET denominada “*id*” puede hacer referencia a un registro de una tabla donde se guarda la información del documento que demanda el usuario.

Otras posibilidades consisten en combinar el identificador, con otros datos, como el título, autor o fecha de publicación de cada documento. Estos datos y valores combinados, ayudan a generar una URI (Uniform Resource Identifier), o lo que es lo mismo, direcciones URL unívocas que identifican cada recurso documental del sitio web o del sistema de información.

```

<!-- Tipo de enlace: ejemplos -->
http://wapache/index.php?page=document001
http://wapache/index.php?page=titulo-del-documento
http://wapache/index.php?page=titulo-del-documento&date=2011-12-15

<?php
// Se establece como primera condición que la variable "page"
// con la que se conforma la URL original, sea distinta a un
// contenido vacío ó se encuentre en blanco. En tal caso, no
// se enlaza ni redirecciona ningún contenido
if($_GET[page] != ""){

    // CASO 1 =====
    // Si el origen de la página citante es "index" o lo que
    // es lo mismo, la página de portada del sitio web, se
    // genera una URL ( documents/[nombre-pagina].php )
    if($origin == "index"){

        // Obsérvese que la función de impresión en pantalla
        // contiene un código Javascript que permite
        // redireccionar a una página web predeterminada
        echo "
        <script language='javascript'>
        window.location.href='documents/$_GET[page].php'
        </script>
        ";

    } // Fin de caso 1

    // CASO 2 =====
    // Si el origen de la página citante es otro documento que
    // se encuentra en la misma carpeta "documents", la URL
    // relativa a generar sólo requerirá el nombre del archivo
    // almacenado en la variable "page"
    if($origin == "doc"){

        // Se repite el código javascript con distinta URL
        // a la que debe ser redireccionado
        echo "
        <script language='javascript'>
        window.location.href='$_GET[page].php'
        </script>
        ";

    } // Fin de caso 2

} else {}
?>

```

Tabla 35. Gestor de enlazamiento y redirección basado en variables GET

Por otro lado, el ejemplo de la tabla 36, muestra un gestor de enlazamiento y redireccionamiento especializado en enlaces de tipo F, caracterizados por no disponer de extensión alguna del tipo de archivo, así como por constar de un nombre completo optimizado para su lectura y recuperación por buscadores y usuarios “*friendly links*”. La forma de operar es distinta en cuanto a la forma de interpretar la información del enlace. En este caso no se recibe el nombre del archivo citado por método GET, de hecho no hay variables que se envíen a través de la URL.

Por ese motivo es necesario captar esa información a través de la función y variable de servidor `pathinfo("$_SERVER[REQUEST_URI]");` que extrae el enlace entrante en el que hizo clic el usuario. Permite obtener toda la información de directorios, subdirectorios y nombre de archivo base que corresponde al documento citado en el enlace. A este recurso "citado", se le denomina también "basename" argumento por el que se extrae su nombre. Una vez que es conocido, se pueden construir los distintos casos y condicionantes, por ejemplo; en el caso de que "basename" no sea la propia página de portada "index.php" se estará refiriendo a un objeto o elemento distinto, es decir, un documento y en tal caso el usuario pretende acceder a él. Esto implica a su vez tener en cuenta el origen del enlace, para lo que se comprueba el origen de la página citante y se redirecciona en función del caso a la página citada referida en la variable `$path_parts[basename]` añadiendo la extensión de tipo de archivo que corresponda.

```

<!-- Tipo de enlace: ejemplos -->
http://wapache/document001
http://wapache/titulo-del-documento-propiamente-dicho
http://wapache/2011-12-15/titulo-del-documento

<?php
// La variable "$path_parts" almacena la URL completa del
// enlace original. Al emplear la función "pathinfo"
// se pueden distinguir todos los directorios, subdirectorios
// y nombre del archivo de base ó "basename" de la URL
$path_parts = pathinfo($_SERVER[REQUEST_URI]);

// CASO 1 =====
// Dado que toda la información de la URL original está
// almacenada, se consulta el nombre del archivo de base.
// Si coincide con la página index, de entrada del
// sitio web, significará que la solicitud del usuario es una
// redirección ó actualización de la página de portada.
// Debido a que el enlace a la portada es único se opta por
// dejar vacía la acción de ejecución del caso, pero podrían
// añadirse procesos de redireccionamiento exprefeso
if($path_parts[basename] == "index.php") { }

// CASO 2 =====
// Si el nombre del archivo de base es distinto a "index.php"
// se ejecuta el caso2, comprobándose que el origen de la
// página citante es la portada del sitio. De esta forma se
// compone la URL a medida con la ruta de directorios
// correspondiente hasta el archivo demandado
elseif($origin == "index") {
    echo "
    <script language='javascript'>
    window.location.href='../documents/$path_parts[basename].php'
    </script>
    ";
}

```

```
// CASO 3 =====
// Si las condiciones del caso1 y 2 no se cumplen, se comprueba
// que el origen de la página citante es otro documento, al
// que se aplica una composición URL distinta
elseif($origin == "doc") {
    echo "
    <script language='javascript'>
    window.location.href='../$path_parts[basename].php'
    </script>
    ";
}

// CASO 4 =====
// Si ninguno de los casos anteriores coincide con la demanda
// se determina que no se efectúe redirección alguna
else {}

?>
```

Tabla 36. Gestor de enlazamiento y redirección para modelo-F de enlaces

18. Ejercicios prácticos

- Práctica 1. Deconstrucción de un sitio web
- Práctica 2. Planificación y desarrollo del sitio web: primeros pasos
- Práctica 3. Editando las páginas de contenidos
- Práctica 4. Construyendo la estructura del sitio web
- Práctica 5. Adaptación dinámica de las páginas de contenidos
- Práctica 6. Controlando la información de los formularios
- Práctica 7. Creando la trama de navegación
- Práctica 8. Gestión, redirección y construcción de enlaces
- Práctica 9. Implementación de un buscador básico
- Práctica 10. Test de usabilidad

Práctica1. Deconstrucción de un sitio web

Teniendo en cuenta qué método de trabajo se utiliza a la hora de crear un proyecto de arquitectura de la información para un documento o sitio web, trata de efectuar la operación inversa. Selecciona una de las siguientes páginas web de tipo académico y trata de generar un bosquejo que represente la estructura y elementos del sitio en su ubicación espacial, un esquema de sus apartados, secciones y principales contenidos, una muestra de lenguaje controlado utilizado en el sitio web y una relación de los metadatos empleados en las páginas principales con un nivel de profundidad.

Lista de sitios web académicos	
http://www.harvard.edu/	http://www.ubc.ca/
http://www.mit.edu/	http://www.gatech.edu/
http://www.stanford.edu/	http://www.ethz.ch/
http://www.berkeley.edu/	http://www.ox.ac.uk/
http://www.cornell.edu/	http://www.osu.edu/
http://www.umich.edu/	http://www.uchicago.edu/
http://www.umn.edu/	http://www.usc.edu/
http://www.washington.edu/	http://www.uga.edu/
http://www.wisc.edu/	http://www.ucdavis.edu/
http://www.utexas.edu/	http://www.uci.edu/
http://www.upenn.edu/	http://www.iastate.edu/
http://www.psu.edu/	http://www.colorado.edu/
http://www.columbia.edu/	http://www.uconn.edu/
http://www.cmu.edu/	http://www.u-tokyo.ac.jp/
http://www.uiuc.edu/	http://www.ed.ac.uk/
http://www.ucla.edu/	http://www.uio.no/
http://www.tamu.edu/	http://oregonstate.edu/
http://www.umd.edu/	http://www.jhu.edu/
http://www.purdue.edu/	http://www.asu.edu/
http://www.unc.edu/	http://www.ucsb.edu/
http://www.msu.edu/	http://www.utah.edu/

http://www.cam.ac.uk/ http://www.ufl.edu/ http://www.rutgers.edu/ http://www.indiana.edu/ http://www.nyu.edu/ http://www.ncsu.edu/ http://www.virginia.edu/ http://www.yale.edu/ http://www.arizona.edu/ http://www.utoronto.ca/ http://www.ucsd.edu/ http://www.pitt.edu/ http://www.princeton.edu/ http://www.duke.edu/ http://www.caltech.edu/ http://www.vt.edu/	http://www.helsinki.fi/ http://www.umontreal.ca/ http://www.ntnu.no/ http://www.uiowa.edu/ http://www.ucl.ac.uk/ http://www.wustl.edu/ http://www.unl.edu/ http://www.sfu.ca/ http://www.gmu.edu/ http://www.ualberta.ca/ http://www.univie.ac.at/ http://www.unam.mx/ http://www.bu.edu/ http://www.epfl.ch/ http://www.anu.edu.au/ http://umass.edu/
--	--

1. ¿Qué sitio web seleccionaste?

Indica el nombre de la universidad y enlace del dominio que has seleccionado

2. Bosquejo de la estructura del sitio web elegido.

Deberá ser perceptible la estructura de marcos, encabezamientos, título, buscador, pie de página, menú lateral, menú horizontal, menú contextual o de navegación, logotipo, secciones de contenidos de primera página, noticias o últimas publicaciones, páginas de presentación, área de contenidos, área de sindicación y comunicación social, datos de contacto y filiación. Copia su estructura elaborando un bosquejo y pega la imagen resultante en el siguiente espacio.

3. Diagrama del sitio web elegido.

Elabora un diagrama con la estructura principal del sitio a 1 nivel de profundidad. Aprovecha los mapas de sitio web disponibles, comprueba las secciones o áreas temáticas de que dispone el sitio para organizar su contenido y plásmalo en un diagrama.

4. Muestra de lenguaje controlado.

Extraer los términos que componen el título de las secciones o apartados del sitio web y sus descripciones evitando palabras vacías, manteniendo sustantivos, evitando verbos y

artículos determinados e indeterminados. A continuación trata de ordenarlos jerárquicamente estableciendo relaciones genéricas y específicas entre ellos. En caso de faltar términos para elaborar el cuadro jerárquico, introduzca aquellos que fueren necesarios.

Lista de términos	Organización jerárquica

5. Metadatos del sitio web.

Extraer todos los metadatos del sitio web. Analiza el código fuente del sitio web y reseña en los siguientes apartados los metadatos y las metaetiquetas de que dispone.

Metadatos Dublin Core
Metaetiquetas HTML

Práctica2. Planificación y desarrollo del sitio web

Para el correcto desarrollo de las prácticas, se utilizarán principalmente dos editores web. Por un lado “*Dreamweaver*” y por otro “*Eclipse*”. El primero está pensado para la edición de sitios web a partir de un interfaz gráfico de opciones y herramientas que permiten escribir el código sin un conocimiento profundo en HTML, CSS, Javascript, etc. El segundo implica un conocimiento más exhaustivo de la programación web, ya que no dispone de asistencia a base de plantillas con código pre-programado, tal como sucedería con Dreamweaver. El mejor editor para trabajar edición web es Eclipse dada su capacidad para adaptarse a la programación en cualquier lenguaje y a la simplicidad de su método de trabajo y soporte. En cambio Dreamweaver se recomienda al comienzo del proceso de aprendizaje de la edición web.

- *Macromedia Dreamweaver* [<http://www.adobe.com/es/products/dreamweaver.html>]
- *Eclipse PDT* [<http://www.eclipse.org/>]

Para resolver las primeras prácticas se propondrá el uso de la herramienta Dreamweaver, de tal forma que se adquiriera el método de edición y la experiencia necesaria antes de probar el editor Eclipse. En ambos casos constituyen herramientas de edición y construcción completa de un sitio web.

Otras herramientas importantes

A parte de los editores web con los que se articula toda la construcción web, su programación y código, se recomienda el empleo conjunto de otras herramientas de diseño como por ejemplo “*Corel Paint Shop Pro*”, “*Fireworks*” o “*Photoshop*”, que permiten la edición y diseño de gráficos para el sitio web. Por otro lado herramientas para elaborar paletas de colores normalizadas para todo el sitio web, como por ejemplo “*Color Code Tool*”.

- *PaintShopPro* [<http://www.corel.com/servlet/Satellite/ca/en/Product/1150981051301>]
- *Macromedia Fireworks* [<http://www.adobe.com/es/products/fireworks.html>]
- *Adobe Photoshop* [<http://www.adobe.com/es/products/photoshop.html>]
- *Colour Lovers Palette Editor* [<http://www.colourlovers.com/palettes/add>]
- *HTML Color Code Tool* [<http://www.computerhope.com/cgi-bin/htmlcolor.pl>]

Teniendo en cuenta las consideraciones previas, seleccionar un área de trabajo, recabar información y determinar las principales áreas de contenido del futuro sitio web.

1. Áreas de trabajo. Marcar área de trabajo objetivo para la planificación y desarrollo del sitio web.	
Biblioteconomía y Documentación en Italia	
Fuentes de información en Biomedicina	
Fuentes de información en Tecnologías de la Documentación <ul style="list-style-type: none"> – Programas para la gestión de bibliotecas, archivos y museos – Programas CMS – Programas para la gestión de fotografías 	
Bibliotecas Universitarias	
Fuentes de información en Ciencia y Tecnología <ul style="list-style-type: none"> – Fuentes especializadas en Matemáticas – Fuentes especializadas en Física – Fuentes especializadas en Química – Fuentes especializadas en Geografía – Fuentes especializadas en Economía 	
Fuentes de información en Humanidades y Ciencias Sociales <ul style="list-style-type: none"> – Fuentes especializadas en pedagogía – Fuentes especializadas en psicología – Fuentes especializadas en sociología – Fuentes especializadas en ciencias jurídicas 	
Bibliotecas Públicas	
Archivos Estatales	
Bases de datos	
Blogs de biblioteconomía y documentación	
Libro electrónico	
Documentación y redes sociales	
Universidades	
Centros de investigación	
Escuelas de negocios	
Repositorios digitales	

2. Búsqueda bibliográfica. Recopilar información pertinente sobre el tema o área de trabajo asignada para organizar al menos 4 secciones de contenido principales. Reseñar 10 fuentes de información con URL que se utilizarán para extraer el corpus de contenidos del futuro sitio web.

3. Secciones principales del sitio web. A partir de la información recopilada reseñar las secciones o categorías principales del sitio web. Al menos 4.

4. Estructura jerárquica de contenidos. A partir de las secciones especificadas anteriormente, crear una clasificación jerárquica con 1 nivel de profundidad con las sub-categorías correspondientes. Al menos 3 o 4 sub-categorías por sección.

--

5. Seleccionar contenidos para sub-categorías. Para cada sub-categoría se seleccionarán textos y párrafos con contenidos obtenidos en el paso 2 de búsqueda bibliográfica de tal forma que queden reseñados los textos e informaciones que se utilizarán para cada apartado. La extensión tendrá que ser equivalente a una página de texto de Word.

Título de la sub-categoría: [reseñar título] – [categoría madre]

Contenido: [pegar contenido]

Imágenes: [pegar contenido]

6. **Diseñar vocabulario controlado para cada categoría.** Crear un vocabulario de 10 términos por sección, que será utilizado a lo largo del trabajo.

Sección 1 (término1, termino2, ...)

Sección 2 (término1, termino2, ...)

Sección 3 (término1, termino2, ...)

Sección 4 (término1, termino2, ...)

7. **Diseñar bosquejo de sitio web.** Dibujar un bosquejo de la estructura principal del sitio web de tal forma, que dicho esquema se repita a lo largo de la navegación de todos sus contenidos.

[Pegar imagen del bosquejo]

8. **Impresión y presentación del informe.** Se deberá imprimir y presentar el presente informe al profesor para su visto bueno. Se recomienda buena presentación, estilo y corrección. Esta práctica es clave para el correcto desarrollo de las sucesivas.

Práctica3. Editando las páginas de contenidos

Para cada sub-categoría de contenidos definida en la práctica2, se deberá elaborar una página web simple en formato HTML y CSS teniendo en cuenta los principios expuestos en los artículos de maquetación de textos y estructuración de páginas de contenidos. Recuérdese que el objetivo es la elaboración de páginas de contenidos lo más sencillas posibles, sin artificios, con calidad para la lectura de los textos, máxima visibilidad, legibilidad y correcta maquetación, control de márgenes, entre otros. Comprobar que las páginas cumplen en todo caso con el siguiente pliego de condiciones.

Aspectos relativos a la estructura

- Encabezado con el título de la página (obligatorio).
- Resumen (opcional).
- Autor (obligatorio).
- Versión (obligatorio).
- Derechos (opcional).
- Tabla de contenidos (obligatorio).
- Cuerpo con contenidos (obligatorio).
- Fuentes y bibliografía (obligatorio).

Aspectos relativos a la maquetación y visualización de la información

- Fondo de página uniforme con color “#ffffff”.
- Estilo de enlaces predeterminado (No requerirá estilo).
- Ancho de la página “100%”.
- Aplicar márgenes externos e internos a todo el conjunto del documento.
- El título de cada apartado o epígrafe deberá quedar resaltado sobre el texto de forma clara y limpia.
- Todos los elementos textuales e imágenes deberán estar encapsulados con capas “<div>” y dispondrán de estilos comunes para su representación. Por ejemplo imágenes con capa y color de fondo grisáceo, borde redondo con grosor de “2px” de tipo sólido y color gris ligeramente más oscuro para destacar sobre el conjunto.

- Textos justificados, excepto aquellos párrafos cortos o elementos de una enumeración o listado que no cubran el ancho del sitio web.
- La tabla de contenidos permitirá enlazar cada uno de los apartados del contenido mediante enlaces anchor para distintas áreas del documento.
- El texto y contenidos deberán contener un sistema de navegación hipertextual con referencias a recursos web externos. Por lo menos 10 enlaces de navegación hipertextual.
- Las fuentes y bibliografía utilizadas deberán estar referenciadas en estilo “*HARVARD Anglia*” y aparecerán reseñadas como una lista “” a base de puntos. Aplicar estilo oportuno.

Práctica4. Construyendo la estructura del sitio web

De acuerdo a lo que se viene explicando, uno de los métodos más efectivos para lograr una estructura de sitio web es el empleo de funciones include de PHP para introducir los distintos fragmentos que componen el código del sitio web. En la práctica4 se construirá la estructura del sitio web conforme a este método. Aunque el editor Eclipse PDT, permite la edición del lenguaje PHP, será necesario utilizar un servidor Apache HTTP y un intérprete del lenguaje PHP para poder comprobar los resultados de la programación efectuada. Se recomienda utilizar una distribución AMP portable como AMPdoc 2.0, disponible en: <http://mblazquez.es/ampdoc-2-0/>

1. Teniendo en cuenta el bosquejo de sitio web definido en la práctica2, crea la estructura de áreas mediante capas “<div>”. Copiar y pegar el código completo de la estructura HTML en el siguiente recuadro.

2. A partir del código fuente que se ha introducido en el recuadro anterior, descomponerlo en archivos PHP utilizando la técnica include explicada en clase. Copia y pega el código fuente utilizado para cada archivo.

Archivo matriz: index.php

Archivo auxiliar: inc.head.php

Archivo auxiliar: inc.menu1.php

Archivo auxiliar: inc.menu2.php

Archivo auxiliar: inc.foot.php

3. Prueba la estructura del sitio web PHP con la distribución AMP. Imprimir pantalla de la prueba efectuada.

Práctica5. Adaptación dinámica de una página de contenidos

Teniendo en cuenta las páginas de contenidos elaboradas en la práctica3, adaptar su contenido a la estructura del sitio web PHP que fue creada en la práctica4. Utiliza el siguiente ejemplo para realizar la práctica:

- **Descargar ejemplo de la práctica 5.** Disponible en:

<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/documentos/practica5-ejemplo.zip>

1. Toma una de las páginas de contenido creadas en la práctica3 y pega su código fuente en el siguiente recuadro.

2. Modifica la página de contenidos de tal manera que se convierta en un archivo matriz como en el caso de la práctica4, para que mantenga el mismo encabezamiento, menús y pie de página que la estructura general del sitio web. Pegar el código fuente resultante de esta transformación en el siguiente recuadro.

Archivo matriz: pagina-de-contenido.php

3. Prueba la estructura de la página de contenidos recién editada con la distribución AMP. Imprimir pantalla de la prueba efectuada.

Práctica6. Controlando la información de los formularios

1. Crea un formulario de contacto con los elementos “*input*”, “*textarea*”, “*radio*”, “*checkbox*”, y “*select*”. Dentro de la misma página del formulario crea un código PHP para procesar la información y presentarla en pantalla. Emplea los conocimientos aprendidos sobre arquitectura de la información y visualización.

[Pega aquí el código fuente]

[Impresión de pantalla]

2. Crea un archivo llamado “*punto1.html*” con el siguiente enlace:

```
punto2.php?identificador=10&url1=biblioteca.php&url2=archivo.php&url3=museo.php&title=fuentes%20documentales&texto=El%20tratamiento%20de%20las%20fuentes%20documentales,%20si%20pretende%20ser%20científico,%20tiene%20que%20partir%20de%20una%20crítica%20de%20las%20fuentes,%20es%20decir,%20del%20juicio%20que%20el%20investigador
```

3. Crea un archivo llamado “*punto2.php*” para procesar la información del enlace y presentarla en pantalla. Emplea los conocimientos aprendidos sobre arquitectura de la información y visualización.

[Pega aquí el código fuente]

[Impresión de pantalla]

Práctica7. Creando la trama de navegación

- **Descargar ejemplo de la práctica 7.** Disponible en:

<http://www.mblazquez.es/blog-ccdodoc-arquitectura-informacion/documentos/practica7-ejemplo.zip>

1. Comprueba el funcionamiento del sitio web de ejemplo de la práctica7.

[Pegar impresión de pantalla]

2. Crea las 4 páginas para las distintas secciones definidas para el proyecto de sitio web. Estas páginas deberán contener una breve descripción de los contenidos adscritos a ellas. Por otro lado añade un breve listado con los enlaces de cada sección, siguiendo para todo el conjunto los principios de arquitectura establecidos.

Archivo documents/section1.php

[Pegar código fuente]

Archivo documents/section2.php

[Pegar código fuente]

Archivo documents/section3.php

[Pegar código fuente]

Archivo documents/section4.php

[Pegar código fuente]

3. Adapta el sistema de navegación dinámico a la estructura definida para el proyecto de tu sitio web, teniendo en cuenta que los enlaces del menú general deberán ser exclusivamente de las páginas de cada sección y que los enlaces del menú local deberán incluir además los enlaces a las páginas de contenidos.

Archivo inc.menu1.php: Código fuente del menú general

[Pegar código fuente]

Archivo inc.menu2.php: Código fuente del menú local

[Pegar código fuente]

[Pegar impresión de pantalla]

4. Una vez elaborados los menús con sus correspondientes enlaces dinámicos, aplicar estilos de enlaces y menús con CSS (Reseñar código). Con el resultado obtenido imprimir pantalla.

Código CSS utilizado

[Pegar código fuente]

[Pegar impresión de pantalla]

Práctica8. Gestión, redirección y construcción de enlaces

- **Descargar ejemplo de la práctica 8.** Disponible en:

<http://www.mblazquez.es/blog-ccdodoc-arquitectura-informacion/documentos/practica8-ejemplo.zip>

1. Comprueba el funcionamiento del sitio web de ejemplo de la práctica8.

[Pegar impresión de pantalla]

2. Copia y pega la URL de los enlaces del menú.

- ¿Podría funcionar el enlazamiento si se suprime del código fuente del menú “*inc.menu1.php*” la denominación “*index.php*”? Justifica y razona tu respuesta.

--

3. Prueba el siguiente código fuente en AMP y responde a los siguientes apartados de la tabla.

Código fuente
<pre><?php \$path_parts = pathinfo(\$_SERVER[REQUEST_URI]); echo "Ruta parte1: \$path_parts[dirname]
"; echo "Ruta parte2: \$path_parts[basename]
"; echo "Ruta parte3: \$path_parts[extension]
"; echo "Ruta parte4: \$path_parts[filename]
"; ¿></pre>
[Pegar impresión de pantalla]

Ruta	Resultado	Corresponde a...
Parte1 – dirname		
Parte2 – basename		
Parte3 – extension		
Parte4 – filename		

4. Toma la iniciativa y cita qué otras 4 funciones de PHP pueden ayudar en la obtención de información de las rutas de enlaces. Para ello utiliza el recurso <http://www.php.net>

Función	Describe utilidad o finalidad de la función

Práctica9. Implementación de un buscador básico

- **Descargar ejemplo de la práctica 9.** Disponible en:

<http://www.mblazquez.es/blog-ccdod-arquitectura-informacion/documentos/practica9-ejemplo.zip>

1. Comprueba el funcionamiento del sitio web de ejemplo de la práctica9.

[Pegar impresión de pantalla]

2. Se observará un nuevo elemento en el sitio web, se trata de un buscador básico con una caja de consulta, en el área del menú general “*inc.menu1.php*”. El buscador actúa sobre todos los documentos alojados en la carpeta “*documents*”, comparando la cadena de búsqueda del usuario y el texto de la base de conocimiento. Para comprobar el funcionamiento del buscador se proponen las siguientes búsquedas:

Nº	Consulta	Resultados (nombre/s de archivo)
1	biblioteconomía	
2	weblog	
3	telev	
4	Pedro Pérez Albéniz	
5	Voz bibliografía	
6	Classificazione delle risorse	
7	Hinari	

- Responde a las siguientes preguntas:

¿Se encontraron resultados pertinentes para cada consulta?

¿Qué términos o palabras coinciden con “telev” en la página de resultados? ¿Significa que el buscador, recupera los contenidos, incluso por fragmentos de palabras?

¿Si se repita la consulta nº4 sin mayúsculas, con y sin acentos, se recupera el mismo documento? ¿a qué puede ser debido? Justifica tu respuesta

3. Como se puede comprobar, el buscador funciona correctamente para efectuar consultas sencillas. Pero su diseño y concepción básica, actúa con grandes limitaciones que son identificativas y académicamente divulgativas de la importancia de utilizar técnicas de recuperación de información. Realiza las siguientes consultas.

Nº	Consulta	Resultados (nombre/s de archivo)
8	biblioteca	
9	bi	
10	tel	
11	y	
12	de	
13	Classificazione risorse	

- Responde a las siguientes preguntas:

La consulta nº8 recupera tres resultados, pero ¿cuáles son los términos coincidentes? ¿Tienen distintas desinencias? ¿Se podría decir que el modelo de recuperación actuó correctamente?

¿Qué términos o palabras coinciden con “telev” en la página de resultados? ¿Significa que el buscador, recupera los contenidos, incluso por fragmentos de palabras?

¿La consulta nº10 produce los mismos resultados que la consulta nº3? De entre los resultados ¿cuáles son los términos coincidentes? ¿Son pertinentes los resultados obtenidos con respecto a la temática del documento3?

Las consultas nº11 y 12 hacen alusión a una conjunción y una preposición. Ambas son consideradas palabras vacías y sin embargo permiten recuperar términos. ¿Responde el buscador en este apartado a los principios de la recuperación de información? ¿Cómo lo arreglarías?

¿En qué se distingue la consulta nº13 de la nº6? ¿Qué falla? ¿No se debería recuperar la misma información que en la consulta nº6?

A tenor de todo lo experimentado, ¿qué tipo de sistema de recuperación (algoritmo) se está utilizando?
Pista... (<http://ccdoc-arquitecturainformacionweb.blogspot.com/2011/10/06-sistemas-de-busqueda.html>)

4. Cómo mejorar el buscador; parte1. Sugiere 5 ideas para hacer más eficaz el sistema, según los principios de arquitectura de la información y representación de los contenidos.

- Pista: <http://ccdoc-arquitecturainformacionweb.blogspot.com/2011/10/08-representacion-de-resultados-e.html>

1	
2	
3	
4	
5	

5. Cómo mejorar el buscador: parte2. Sugiere 5 ideas para hacer más eficaz el sistema, en cuanto a procesos de tratamiento de los textos y cadenas de consulta del usuario.

- Pista: <http://ccdoc-arquitecturainformacionweb.blogspot.com/2011/10/06-sistemas-de-busqueda.html>

1	
2	
3	
4	
5	

6. Modifica el formato y características de la página de resultados de acuerdo a los principios de arquitectura de la información


Reseña el código de la página de resultados y CSS implicado

[Pegar impresión de pantalla]

Práctica10. Test de usabilidad

Uno de los enfoques de la usabilidad corresponde al comportamiento del usuario durante el proceso de navegación, hasta la localización de los contenidos que resuelven su demanda informativa. Para experimentar el número de clics que el usuario realiza durante una búsqueda y evaluar la usabilidad de los menús de navegación global, local y contextual se pueden llevar a cabo test de usabilidad. En el caso propuesto, se presenta un test experimental desarrollado exprofeso para la asignatura, en el que se indican a lo largo de 10 preguntas distintas consultas que el usuario deberá resolver en 1 minuto. Cada movimiento y enlace es recopilado de forma tal que al finalizar el test puede ser analizado y estudiado desde un punto de vista cualitativo.

Universidad Complutense de Madrid
Facultad de Ciencias de la Documentación




Test de usabilidad de un medio de comunicación
Prof. Manuel Blázquez Ochando

Instrucciones: El presente test de usabilidad tiene como objetivo servir de prueba para la recopilación dinámica de datos que reflejen el comportamiento real del usuario en un proceso de navegación y consulta de información. A lo largo del test se irán formulando diversas preguntas en las que se propone al usuario encontrar una información sito en algún punto indeterminado del sitio web objetivo. Para responder a las preguntas tiene a su disposición un marco con la página objetivo en la que deberá navegar normalmente hasta dar con la respuesta que considere correcta. Para hacer esta operación contará con un tiempo limitado. El siguiente bloque de preguntas, versará sobre aspectos propios de la usabilidad de la página, por ejemplo localización de la página o sitio en el que se encuentra el usuario, secciones del sitio, opciones del nivel en el que se encuentra el usuario, etc.

[Comenzar Test](#)

Copyright 2012 - Todos los derechos reservados



Prueba de un test de usabilidad de un medio de comunicación. Disponible en:
<http://www.mblazquez.es/blog-ccdoc-arquitectura-informacion/test-usabilidad1/test1.php>

19. Bibliografía y referencias

- MARTÍN FERNÁNDEZ, F.J.; HASSAN MONTERO, Y. 2003. Qué es la Arquitectura de la Información. Disponible en: <http://www.nosolousabilidad.com/articulos/ai.htm>
- HASSAN MONTERO, Y. 2002. Cómo leen los usuarios en la Web. Disponible en: http://www.nosolousabilidad.com/articulos/como_leen_usuarios.htm
- HASSAN, Y.; MARTÍN FERNÁNDEZ, F.J.; IAZZA, G. 2004. Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información. En Hipertext. Disponible en: <http://www.hipertext.net/web/pag206.htm>
- GARRET, J.J. 2011. The elements of user experience: user-centered design for the web and beyond. New Rider
- HASSAN, Y.; MARTÍN FERNÁNDEZ, F.J.; GHZALA, I. 2004. Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información. Hipertext, n.2. Disponible en: <http://www.hipertext.net/web/pag206.htm>
- HOM, J. 1998. The Usability Methods Toolbox Handbook. Disponible en: <http://www.idemployee.id.tue.nl/g.w.m.rauterberg/lecturenotes/UsabilityMethodsToolboxHandbook.pdf>
- KRUG, S. 2002. Don't make me think: a common sense approach to web usability = No me hagas pensar: Una aproximación a la usabilidad en la web. Prentice Hall
- NIELSEN, J.; LORANGER, H. 2006. Prioritizing Web Usability. New Riders
- FLORÍA CORTÉS, A. 2000. Recopilación de Métodos de Usabilidad. Disponible en: <http://www.sidar.org/recur/desdi/traduc/es/visitable/Herramientas.htm>
- W3C. 2000. HTML4.01: Iframe specs. Disponible en: <http://www.w3.org/TR/REC-html40/present/frames.html#edef-IFRAME>
- PHP Group. 2011. Include. Disponible en: <http://php.net/manual/es/function.include.php>
- W3.org. 2015. HTTP/1.1: Method Definitions. Part of Hypertext Transfer Protocol. Disponible en: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>
- PHP Group. 2011. \$_SERVER[] Server and execution environment information. Disponible en: <http://www.php.net/manual/en/reserved.variables.server.php>
- PHP Group. 2011. pathinfo() Information about a file path. Disponible en: <http://php.net/manual/es/function.pathinfo.php>

20. Índice de tablas

Tabla 1. Estructura web basada en marcos o frames	38
Tabla 2. Estructura web basada en capas	39
Tabla 3. Estructura web basada en capas e iframe	40
Tabla 4. Enlace en HTML con todos sus atributos.....	46
Tabla 5. Estilo CSS para los enlaces HTML	46
Tabla 6. Modelo1 de enlaces	47
Tabla 7. Diferenciando dos estilos de enlaces	47
Tabla 8. Modelo 2 de enlaces	49
Tabla 9. Modelo de menú horizontal.....	50
Tabla 10. Modelo de menú desplegable vertical	51
Tabla 11. Modelo de menú lateral	52
Tabla 12. Vinculación de archivos CSS	53
Tabla 13. Propiedades del texto.....	57
Tabla 14. Encapsulado de un texto en una capa	58
Tabla 15. Ejemplo de propiedades de las capas	58
Tabla 16. Márgenes internos y externos	58
Tabla 17. Alineación de textos a derecha, izquierda, centro o justificado	59
Tabla 18. Sangrado de textos.....	59
Tabla 19. Dimensiones de las capas	59
Tabla 20. Ejemplo de integración de imágenes y texto dentro de la misma capa.....	60
Tabla 21. Ejemplo de imágenes y textos en capas independientes	60
Tabla 22. Aplicación de estilos en listas ordenadas y desordenadas.....	61
Tabla 23. Sitio web con método iframe para cargar páginas y contenidos	63
Tabla 24. Ejecución en el navegador del código fuente de la tabla 23.....	64
Tabla 25. Ejemplo de aplicación del método include para componer sitios web.....	66
Tabla 26. Ejecución en el navegador del código de la tabla 25.....	66
Tabla 27. Ejemplo de adaptación dinámica de una página con el método include	70
Tabla 28. Principales componentes del formulario	72
Tabla 29. Descripción de componentes del formulario.....	73
Tabla 30. Formulario para el envío de datos con método POST.....	74
Tabla 31. Archivo de recepción que presenta en pantalla la información del mensaje..	75
Tabla 32. Archivo de envío y recepción de datos.....	75

Tabla 33. Archivo de envío y recepción de datos.....	76
Tabla 34. Tipología de enlaces de la web.....	79
Tabla 35. Gestor de enlazamiento y redirección basado en variables GET	82
Tabla 36. Gestor de enlazamiento y redirección para modelo-F de enlaces	84

21. Índice de figuras

Figura 1. La Arquitectura de la Información en el centro de los servicios documentales	4
Figura 2. Diferentes entornos de la Arquitectura de la Información	5
Figura 3. Cómo la experiencia del usuario transforma los servicios de información	8
Figura 4. Muestra de estructura básica de un sitio web estático.....	54
Figura 5. Muestra de estructura básica de un sitio web dinámico.....	56
Figura 6. Áreas clave de la estructura de un sitio web	62