# Inserción de hipervínculos o enlaces (<A>)

Un hipervínculo también llamado enlace, link, hiperenlace, referencia, ancla, etc., es la utilidad básica del hipertexto y la principal de una página Web. Es un elemento del lenguaje HTML que nos permite acceder otro recurso, pagina Web, imagen, etc. Es el elemento que realmente nos permite "navegar" por Internet.

Un enlace puede ser cualquier elemento del lenguaje HTML como texto, imagen, porción de imagen, ítem de una lista, etc. No hace falta darle ningún aspecto diferenciador para hacerlo reconocible, **los enlaces se destacan automáticamente, cambiando de color y subrayándose en el caso de ser texto o poniendo un cuadro de color a las imágenes**. Además, porque al pasar el puntero del ratón por encima de ellos cambia de aspecto y se convierte en una mano cerrada con el índice apuntando hacia arriba, además en la parte inferior del Navegador aparece información sobre el enlace al que se dirigirá la página cuando hagamos clic sobre él.

La etiqueta que se utiliza para definirlos es

**<A>** texto que hará de enlace**</A>**

También recibe el nombre de **etiqueta de creación de anclas**. Un ancla es una posición de la página Web a la que podemos asignar un nombre.

Un ancla puede actuar como:

* **Origen de un hipervínculo**

Lugar sobre el que tenemos que hacer clic para acceder a otra posición o dirección.

* **Destino de un hipervínculo**

Lugar al que se accede al hacer clic sobre otro hipervínculo.

**NOTA: CONCEPTO DE URI**

*1. Identificar de forma única a ese recurso*

Las URI permiten que cada página HTML publicada en Internet tenga un nombre único que permita diferenciarla de las demás. De esta forma es posible crear enlaces que apunten de forma inequívoca a una determinada página.

*Si se accede a la página principal de Google, la dirección que muestra el navegador es:*

*http://www.google.com Esta es la URI completa de la página principal de Google.*

2. *Permitir localizar de forma eficiente ese recurso*

Para permitir la localización eficiente es necesario comprender las diferentes partes que forman las URI. Una URI sencilla siempre está formada por las mismas tres partes.

*Si por ejemplo se considera la siguiente URI:* [*http://www.elmundo.es/elmundomotor/motos.html*](http://www.elmundo.es/elmundomotor/motos.html)

Las partes que componen la URI anterior son:

 **Protocolo (http://):** el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan http://. Las páginas web SEGURAS (por ejemplo las de los bancos y las de los servicios de email) utilizan https:// (se añade una letra s).

 **Servidor** (www.elmundo.es): simplificando mucho su explicación, se trata del ordenador en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre.

 **Ruta** (/elmundomotor/motos.html): CAMINO que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder.

Por tanto, las URI no sólo identifican de forma única a cada recurso de Internet, sino que también proporcionan a los navegadores la información necesaria para poder llegar hasta ese recurso. La mayoría de URI son tan sencillas como la URI mostrada anteriormente.

No obstante, existen URI complejas formadas por más partes.

[*http://www.alistapart.com/comments/webstandards2008?page=5#42*](http://www.alistapart.com/comments/webstandards2008?page=5#42)

Las cinco partes que forman la URI anterior son:

 Protocolo (http://)

 Servidor (www.alistapart.com)

 Ruta (/comments/webstandards2008)

 Consulta (?page=5): información adicional necesaria para que el servidor localice correctamente el recurso que se quiere acceder. Siempre comienza con el carácter ? y contiene una sucesión de palabras separadas por = y &

 Sección (#42): permite que el navegador se posicione automáticamente en una sección de la página web. Siempre comienza con el carácter #

**Como las URI utilizan los caracteres :, =, & y / para separar sus partes, estos caracteres están reservados**. Además, algunos caracteres no están reservados pero pueden ser problemáticos si se utilizan en la propia URI. (Recomendación RFC 1738 para URL)

Si es necesario incluir estos caracteres reservados y especiales en una URI, se sustituyen por combinaciones de caracteres seguros. Esta sustitución se denomina CODIFICACIÓN de caracteres y el servidor realiza el proceso inverso (DECODIFICACIÓN) cuando le llega una URI con los caracteres codificados.

A continuación se muestra la tabla para codificar los caracteres más comunes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Carácter original** | **Carácter codificado** | **Carácter original** | **Carácter codificado** |
| **ñ** | %F1 | Dolar - $ | %24 |
| **Ñ** | %D1 | Ampersand - & | %26 |
| **á** | %E1 | Más - + | %2B |
| **Á** | %C1 | Coma - , | %2C |
| **é** | %E9 | Slash-/ | %2F |
| **É** | %C9 | Dos Puntos- : | %3A |
| **í** | %ED | Punto y coma- ; | %3B |
| **Í** | %CD | Igual- = | %3D |
| **ó** | %F3 | Interrogación- ? | %3F |
| **Ó** | %D3 | Arroba- @ | %40 |
| **ú** | | %FA | |
| **Ú** | | %DA | |
| **ç** | | %E7 | |
| **Ç** | | %C7 | |

|  |  |
| --- | --- |
| **<!-- URL problemática -->** | **<!-- URL correcta -->** |
| http://www.ejemplo.com/estaciones/otoño.html | http://www.ejemplo.com/estaciones/oto%F1o.html |
| http://www.ejemplo.com/ruta/nombre página.html | http://www.ejemplo.com/ruta/nombre%20p%E1gina.htm |

Para definir los URL en los enlaces se utilizan direcciones absolutas o relativas.

## Tipos de direccionamiento en los enlaces

* **Absoluta**

Se especifica la ruta o camino completo del documento destino, incluso aunque el documento esté en el mismo directorio o en la misma máquina. En función del protocolo de comunicación a establecer podemos encontrarnos con:

* **URL file**

<a href="file:// /unidad:path/fichero">texto</a>

* **URL ftp anónimo**

<a href="ftp://servidor:puerto/path/fichero">texto</a>

* **URL ftp**

<a href="ftp://usuario:password@servidor:puerto/path/fichero; type=code">texto</a>

<a href=”ftp://ftp.uv.es”> Ftp de la Universidad de valencia </a>

* **URL mailto**

<a href="mailto:nombre@misitio.com?subject=Asunto del mensaje">texto</a>

La acción sobre un vínculo “mailto” provocará que se abra un mensaje nuevo de la aplicación de correo por defecto del usuario.

|  |  |
| --- | --- |
| Ejemplo: |  |

El elemento "a" permite direccionar un hipervínculo a un programa de envío de correos que tengamos configurado en nuestra computadora.

Haremos una página que disponga un hipervínculo a un cliente de correo de mail:

<html>

<head>

</head>

<body>

<h1>Reclamos</h1>

<a href="mailto:diegoestevanes@gmail.com">Enviar mail.</a>

</body>

</html>

Cuando se presiona el enlace se abre el programa de envío de correos que tiene configurado el equipo y dispone como receptor del mensaje la dirección que configuramos en el propio enlace seguido de la **palabra mailto**. La sintaxis para poner un título por defecto y un cuerpo de mensaje es:

**<a href="mailto:diegoestevanes@gmail.com?subject=título del mensaje&body=cuerpo del mensaje">Enviar mail.</a>**

**Es decir después de especificar el destinatario del mail ponemos un carácter de interrogación '?' seguido la palabra subject, un igual y el título por defecto que debe aparecer en la ventana de envío de mail. Por último separamos con un ampersand '&' la inicialización de subject y el body (es decir el cuerpo del mensaje)**

Podemos inclusive añadir el envío de mail con copia y con copia oculta a otras direcciones:

<html>

<head>

</head>

<body>

<h1>Reclamaciones</h1>

<a href="mailto:diego1@gmail.com?

subject=aquí el título&cc=diego2@gmail.com&bcc=diego3@gmail.com&body=Este

es el cuerpo">Enviar mail.</a>

</body>

</html>

En este ejemplo enviamos un mail a diego1@gmail.com, con copia a diego2@gmail.com y con copia oculta a diego3@gmail.com

* **URL nntp**

<a href="nntp://servidornews:puerto/gruponews/articulo"> texto</a>

* **URL javascript**

<a href="javascript:codigo\_javascript">texto</a>



Ejemplos:

<a href=”http://www.gogle.es”> Buscador Google </a>

<a href=”http://www.nasa.gov”> La Nasa </a>

<a href=”http://www.vatican.va”> Santa Sede </a>

<a href=http://www.mcu.es/comun/bases/isbn/ISBN.html”> Libros publicados en España </a>

<a href=http://www.fbi.gov/wanted/topten/fugitives/fugitives.htm> Más buscados FBI</a>

* **Relativa**

En este tipo de direcciones se omite la máquina y la ruta del fichero, se entiende que el fichero esta en el mismo directorio, en el directorio indicado con la etiqueta <BASE> en la cabecera o en un directorio distinto que el fichero que contiene el enlace. Si está en la misma máquina pero en otro directorio se encaminará de la siguiente manera:

o **/**

Indica el directorio raíz del servidor de Web, donde está la página principal.

o **/Directorio/fichero.htm**

Indica que el fichero está en un subdirectorio del directorio que contiene el fichero con el enlace.

o **.. /**

Indica que el fichero referenciado en el enlace está en un directorio de nivel superior, esto se puede repetir tantas veces como sea necesario hasta llegar al directorio deseado.

Cuando se utilizan los URL relativos dentro de un sitio Web**, el sitio se vuelve transportable**. Por supuesto, hay un inconveniente potencial en los URL relativos: pueden volverse confusos en un sitio grande, en particular si hay directorios centralizados para cosas como imágenes. Imagina que tiene URLs como “../../imágenes/logo.gif” en archivos profundos en una estructura de sitio. Algunos usuarios pueden estar tentados de simplemente copiar archivos.

Una forma de resolverlo es con el elemento <BASE> que fija un URL absoluto (no puede ser una ruta) que se usará como base para resolver URL relativos dentro de un documento.

<BASE HREF="www.mipagina.com/">

El elemento BASE solo puede aparecer una vez en el documento y sólo se permite dentro de la cabecera de un documento de HTML.

## 1.2. Atributos de los enlaces

Los atributos que tiene la etiqueta **<A>** y que nos permitirán especificar las características de los enlaces son:

* **HREF**

Especificará el destino del enlace. Pueden ser:

* + - Internos

Dirigidos a partes dentro de la misma página. Se debe usar el símbolo “#” para referenciar el punto de la página al que saltarla (Nombrepagina#nombre\_ancla).

* + - Locales

Dirigidos a otros paginas del mismo sitio web.

* + - Remotos

Dirigidos a otras páginas de otros sitios web.

* + - Direcciones de correo

Dirigidos a crear mensajes de correo.

* + - Archivos

Dirigidos a descargas de ficheros.

* **ID(antigua name)**

Asigna un nombre a una posición de la página. Después se podrá utilizar para realizar enlaces internos con esa parte del documento o pagina.

Se podrá acceder a ellos directamente desde otro documento, indicando la URL destino y el nombre del punto de ancla (**URL#nombre\_ancla**).

* **TITLE**

Asigna un texto descriptivo del destino del enlace o de la funcionalidad del mismo, que aparecerá cuando el puntero del ratón se posicione sobre el enlace, en forma del conocido "tip" o bocadillo explicativo, común a diferentes aplicaciones

* **TARGET**

Indicará al navegador dónde debe abrir la nueva página referenciada en el enlace. Puede tomar los valores:

* + - **\_blank**

Indicará que se debe abrir en una ventana nueva.

**Ejemplo :**

El elemento "a" tiene una propiedad **target** que nos permite indicar que la referencia del recurso sea abierta en otra página. A la propiedad target, debemos asignarle el valor **"\_blank"** para indicar que el recurso sea abierto en otra ventana.

Ejemplo: página que contenga dos hipervínculos, el primero abra el sitio en el mismo navegador y el segundo en otra instancia del navegador:

<html>

<head>

</head>

<body>

<h1>Apertura de enlaces en el mismo navegador y en otra instancia del navegador</h1>

<p>

<a href="http://www.elmarca.com">Periódico El marca</a>

<br>

<a href="http://www.Elpais.com" target="\_blank">Periódico

El pais</a>

</p>

</body>

</html>

* + - **\_self**

Indicará que se debe abrir en la misma ventana o frame activo. Se abrirá en la misma página (si la página tiene definidos frames).

* + - **\_parent**

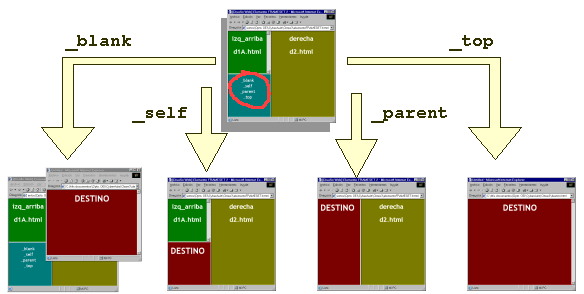
Indicará que se debe abrir en el frame padre o primario.

* + - **\_top**

Indicará que se debe abrir en la misma ventana, ocupando toda la ventana y destruyendo los frames si la página los tuviera.

* + - **Nombre\_ventana**

Indicará que se debe abrir dentro del frame declarado con ese nombre.



<a href="http://www.tutores.org/" target="\_blank">Abre la pagina - Blank </a><br>  
< a href="http://www.tutores.org/" target="\_top">Abre la pagina - Top</a> <br>  
< a href="http://www.tutores.org/" target="\_parent">Abre la pagina - Parent</a> <br>  
< a href="http://www.tutores.org/" target="\_self">Abre la pagina - Self</a><br>  
< a href="http://www.tutores.org/" target="nombre">Abre la pagina - Nombre</a><br>

## 

## 1.3 Hipervínculos a otra página del mismo sitio

**El elemento más importante que tiene una página de internet es el hipervínculo**, estos nos permiten cargar otra página en el navegador. Normalmente un navegador al encontrar esta marca muestra un texto subrayado, y al hacer clic con el ratón sobre éste, el navegador carga la página indicada por dicho hipervínculo. La marca de hipervínculo a otra página del mismo sitio tiene la siguiente sintaxis:

**<a href="pagina2.html">Noticias</a>**

Como vemos, se trata de otro elemento HTML que tiene comienzo de marca y fin de marca. Lo que se encuentra entre el comienzo de marca y el fin de la marca es el texto que aparece en la página (normalmente subrayado).

**Aparece el concepto de propiedad. Una propiedad se incorpora en el comienzo de una marca y tiene un nombre y un valor. El valor de la propiedad debe ir entre comillas dobles**. La propiedad **href** del elemento "a" hace referencia a la página que debe mostrar el navegador si el visitante hace clic sobre el hipervínculo.

Ejemplo: dos páginas que contengan hipervínculos entre sí, los nombres de las páginas HTML serán: pagina1.html y pagina2.html

pagina1.html

<html>

<head>

</head>

<body>

<h1>Página principal.</h1>

<a href="pagina2.html">Noticias</a>

</body>

</html>

pagina2.html

<html>

<head>

</head>

<body>

<h1>Noticias.</h1>

<a href="pagina1.html">Salir.</a>

</body>

</html>

Como podemos observar lo nuevo en la pagina1.html es el hipervínculo a la pagina2.html:

<a href="pagina2.html">Noticias</a>

Toda propiedad toma el valor que se encuentra seguidamente del carácter =

El valor de la propiedad href en este caso es pagina2.html (es otro archivo HTML que debe encontrarse en nuestro sitio y en el mismo directorio).

El segundo archivo pagina2.html tiene un hipervínculo a la primera página:

<a href="pagina1.html">Salir.</a>

Para recordar el nombre de esta marca HTML:

**<a>** viene de **anchor** que significa ancla.

## 1.4 Hipervínculos a otros sitios de Internet

La sintaxis para disponer un hipervínculo a otro sitio de internet es:

**<a href="http://www.google.com">Buscador Google</a>**

**Ahora la propiedad href la inicializamos con el nombre del dominio del otro sitio.**

Es importante tener en cuenta que debemos poner el tipo de protocolo a utilizar. Cuando se trata de una página de internet, el protocolo es el http.

**Resumiendo a la propiedad href la inicializamos con el nombre del protocolo (http) seguida de dos puntos (:) y dos barras (//) luego la cadena (www.) y finalmente el nombre de dominio del sitio a enlazar**.

La siguiente página muestra un hipervínculo al sitio principal del buscador Google:

<html>

<head>

</head>

<body>

<a href="http://www.google.com">Buscador Google</a>

</body>

</html>

Si analizamos un poco y pensamos que esta marca nos permite pedir una página a un servidor para que la cargue en el navegador: Qué página nos retorna del dominio www.google.com ?  
La respuesta es que todo servidor cuando recibe una petición de una página sin indicar su nombre (es decir sólo está el nombre de dominio) selecciona y envía una página que tiene configurada el servidor como página por defecto (generalmente esa página es la principal del sitio y a partir de la cual podemos navegar mediante hipervínculos a otras páginas que se encuentran en dicho dominio).

Podemos enlazar a una página determinada de otro sitio. Veamos un ejemplo, si queremos disponer un enlace (hipervínculo) a la página about.html de google la sintaxis será la siguiente:

<a href="http://www.google.com/intl/en/about.html">Acerca de Google</a>

Debemos conocer exactamente el nombre de la página (en este caso about.html) y también si la página no se encuentra en el directorio raíz debemos saber exactamente el camino de directorios (en este caso /intl/en/)

## 1.5. Enlaces dentro del propio documento

Otra posibilidad que nos ofrece HTML es disponer de una referencia dentro de la página para poder posteriormente poner un hipervínculo a dicha marca.

* El elemento **<a>** tiene dos atributos: **href** y **id**.
* Para la creación de estos hiperenlaces, debemos seguir dos pasos:

1. Marcar las distintas zonas o secciones del documento. Esto lo haremos con el parámetro **id**:

<a **id**= “Nombredeancla”></a>

Parte B del documento

1. Especificar un enlace a cada una de las secciones que hayamos definido.

<a **href**= “#Nombredeancla”> Texto del enlace de ancla </a>

Parte A del documento

Para introducir un vínculo a un recurso concreto de un documento html, se deben definir las anclas de destino mediante el atributo **id** de la etiqueta **<A>**.

**<A id**="nombre\_ancla"**>**texto**</A>**

Para acceder a ese punto, el nombre del destino se indica precedido de **#**.

**<A href="#destino1">**texto**</A>**

La acción sobre el vínculo con un valor **#destino** del atributo **href** situará el destino indicado del documento HTML en la parte superior del navegador.

**Ejemplo de Anclas llamadas desde la misma página.**

Es una práctica común cuando queremos desplazarnos dentro de una página de gran tamaño. Se ponen hipervínculos a diferentes anclas. **La sintaxis para definir un ancla es: <a id="nombreancla"></a>**

No debemos confundir un ancla con un hipervínculo. Para un ancla inicializamos la propiedad **id** con el nombre del ancla. Un ancla se define en una parte de la página que queremos que el operador llegue a partir de un hipervínculo.

Ahora la sintaxis para ir a un ancla desde un hipervínculo es la siguiente:

<a href="#nombreancla">Introducción</a><br>

Vemos que en la propiedad href indicamos el nombre del ancla.

Haremos un ejemplo, donde dispondremos una lista de hipervínculos que llaman a una serie de anclas dispuestas en la misma página:

<html>

<head>

</head>

<body>

<h1>Tutorial de MySQL</h1>

<a href="#introduccion">Introducción</a><br>

<a href="#mostrarbasedatos">show databases</a><br>

<a href="#creaciontabla">Creación de una tabla y mostrar sus campos</a><br>

<a href="#cargarregistros">Carga de registros a una tabla y su recuperación</a><br>

<h2 id="introduccion">Introducción</h2>

<p>

SQL, Structure Query Language (Lenguaje de Consulta Estructurado) es un lenguaje de programacion para trabajar con base de datos relacionales como MySQL, Oracle, etc.<br>

MySQL es un interpretador de SQL, es un servidor de base de datos.<br>

MySQL permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones, etc., resumiendo: administrar bases de datos.<br>

Ingresando instrucciones en la línea de comandos o embebidas en un lenguaje como PHP nos comunicamos con el servidor. Cada sentencia debe acabar con punto y coma (;).<br>

La sensibilidad a mayúsculas y minúsculas, es decir, si hace diferencia entre ellas, depende del sistema operativo, Windows no es sensible, pero Linux

si. Por ejemplo Windows interpreta igualmente las siguientes sentencias:<br>

create database administracion;<br>

Create DataBase administracion;<br>

Pero Linux interpretará como un error la segunda.<br>

Se recomienda usar siempre minúsculas. Es más el sitio mysqlya.com.ar

está instalado sobre un servidor Linux por lo que todos los ejercicios

deberán respetarse mayúsculas y minúsculas.

</p>

<h2 id="mostrarbasedatos">show databases</h2>

<p>

Una base de datos es un conjunto de tablas.<br>

Una base de datos tiene un nombre con el cual accederemos a ella.<br>

Vamos a trabajar en una base de datos ya creada en el sitio, llamada "administracion".<br>

Para que el servidor nos muestre las bases de datos existentes, se lo solicitamos enviando la instrucción:<br>

show databases;<br>

Nos mostrará los nombres de las bases de datos, debe aparecer en este sitio

"administracion".<br>

</p>

<h2 id="creaciontabla">Creación de una tabla y mostrar sus campos</h2>

<p>

Una base de datos almacena sus datos en tablas.<br>

Una tabla es una estructura de datos que organiza los datos en columnas y filas; cada columna es un campo (o atributo) y cada fila, un registro. La intersección de una columna con una fila, contiene un dato específico, un solo valor.<br>

Cada registro contiene un dato por cada columna de la tabla.<br>

Cada campo (columna) debe tener un nombre. El nombre del campo hace referencia a la información que almacenará.<br>

Cada campo (columna) también debe definir el tipo de dato que almacenará.<br>

</p>

<h2 id="cargarregistros">Carga de registros a una tabla y su recuperación</h2>

<p>

Usamos "insert into". Especificamos los nombres de los campos entre paréntesis y separados por comas y luego los valores para cada campo, también entre paréntesis y separados por comas.<br>

Es importante ingresar los valores en el mismo orden en que se nombran los campos, si ingresamos los datos en otro orden, no aparece un mensaje de error y los datos se guardan de modo incorrecto.<br>

Note que los datos ingresados, como corresponden a campos de cadenas de caracteres se colocan entre comillas simples. Las comillas simples son OBLIGATORIAS.

</p>

</body>

</html>

Cada hipervínculo hace referencia a un ancla que se encuentra en la misma página:

<a href="#introduccion">Introducción</a><br>

<a href="#mostrarbasedatos">show databases</a><br>

<a href="#creaciontabla">Creación de una tabla y mostrar sus campos</a><br>

<a href="#cargarregistros">Carga de registros a una tabla y su recuperación</a><br>

Luego la definición de las anclas es:

<a id="introduccion"></a>

<h2>Introducción</h2>

<p>

Como podemos observar la definción del ancla se hace inmediatamente anterior al título donde queremos que el navegador se sitúe.

## 1.6. Enlaces a un determinado lugar de otro documento

Del mismo modo que puede usarse los enlaces internos dentro de un documento, se puede acceder a un recurso concreto de un documento distinto al actual.

Para ellos se incluirá en el atributo **href** el documento que contiene el ancla seguido del carácter # y del identificador del ancla.

**<a href**="pagina.html#parrafo2">texto**</a>**

En **pagina.html**, tendremos un ancla identificada como **"parrafo2"** mediante el atributo **id** comentado previamente.

**Ejemplo de Anclas llamadas desde otra página.**

La llamada a anclas desde otra página (no importa si se encuentra en el mismo sitio o en otro).Debemos conocer el nombre de la página a llamar y el nombre del ancla. La sintaxis para la llamada al ancla es: **<a href="pagina2.html#introduccion">Introducción</a>**

**Es decir después del nombre de la página que llamamos ponemos el carácter # y seguidamente el nombre del ancla.**

Ejemplo: dos páginas y que la primera llame a diferentes anclas definidas en la segunda:

pagina1.html

<html>

<head>

</head>

<body>

<h1>Tutorial de MySQL</h1>

<a href="pagina2.html#introduccion">Introducción</a><br>

<a href="pagina2.html#mostrarbasedatos">show databases</a><br>

<a href="pagina2.html#creaciontabla">Creación de una tabla

y mostrar sus campos</a><br>

<a href="pagina2.html#cargarregistros">Carga de registros a una

tabla y su recuperación</a><br>

</body>

</html>

pagina2.html

<html>

<head>

</head>

<body>

<h2 id="introduccion">Introducción</h2>

<p>

SQL, Structure Query Language (Lenguaje de Consulta Estructurado) es un lenguaje

de programacion para trabajar con base de datos relacionales como MySQL, Oracle,

etc.<br>

MySQL es un interpretador de SQL, es un servidor de base de datos.<br>

MySQL permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones, etc., resumiendo: administrar bases de datos.<br>

Ingresando instrucciones en la linea de comandos o embebidas en un lenguaje como PHP nos comunicamos con el servidor. Cada sentencia debe acabar con punto y coma (;).<br>

La sensibilidad a mayúsculas y minúsculas, es decir, si hace diferencia

entre ellas, depende del sistema operativo, Windows no es sensible, pero Linux

si. Por ejemplo Windows interpreta igualmente las siguientes sentencias:<br>

create database administracion;<br>

Create DataBase administracion;<br>

Pero Linux interpretará como un error la segunda.<br>

Se recomienda usar siempre minúsculas. Es más el sitio mysqlya.com.ar

está instalado sobre un servidor Linux por lo que todos los ejercicios

deberán respetarse mayúsculas y minúsculas.</p>

<a href="pagina1.html">Retornar</a><br>

<h2 id="mostrarbasedatos">show databases</h2>

<p>

Una base de datos es un conjunto de tablas.<br>

Una base de datos tiene un nombre con el cual accederemos a ella.<br>

Vamos a trabajar en una base de datos ya creada en el sitio, llamada "administracion".<br>

Para que el servidor nos muestre las bases de datos existentes, se lo solicitamos enviando la instrucción:<br>

show databases;<br>

Nos mostrará los nombres de las bases de datos, debe aparecer en este sitio

"administracion".<br>

</p>

<a href="pagina1.html">Retornar</a><br>

<h2 id="creaciontabla">Creación de una tabla y mostrar sus campos</h2>

<p>

Una base de datos almacena sus datos en tablas.<br>

Una tabla es una estructura de datos que organiza los datos en columnas y filas; cada columna es un campo (o atributo) y cada fila, un registro. La intersección de una columna con una fila, contiene un dato específico, un solo valor.<br>

Cada registro contiene un dato por cada columna de la tabla.<br>

Cada campo (columna) debe tener un nombre. El nombre del campo hace referencia a la información que almacenará.<br>

Cada campo (columna) también debe definir el tipo de dato que almacenará.<br>

</p>

<a href="pagina1.html">Retornar</a><br>

<h2 id="cargarregistros">Carga de registros a una tabla y su recuperación</h2>

<p>

Usamos "insert into". Especificamos los nombres de los campos entre

paréntesis y separados por comas y luego los valores para cada campo, también entre paréntesis y separados por comas.<br>

Es importante ingresar los valores en el mismo orden en que se nombran los campos, si ingresamos los datos en otro orden, no aparece un mensaje de error y los datos se guardan de modo incorrecto.<br>

Note que los datos ingresados, como corresponden a campos de cadenas de caracteres se colocan entre comillas simples. Las comillas simples son OBLIGATORIAS.

</p>

<a href="pagina1.html">Retornar</a><br>

</body>

</html>

## 1.7. Enlaces con imagen



## 1.8. Colores de los enlaces

Los enlaces en las páginas HTML tienen unas características por defecto que son:

* **Para enlaces con texto**

Color azul y subrayado

* **Para enlaces con imágenes**

La imagen sale con un borde azul, a menos que se sitúe el atributo **BORDER**=0 en la etiqueta **<IMG>**

Una vez activado el enlace su color pasa a ser morado claro.

Todo lo comentado anteriormente es el comportamiento por defecto que tienen los enlaces en un documento HTML, pero se puede cambiar mediante los atributos **link**, **alink** y **vlink** de la etiqueta **<body>**.

## 1.9. Estados de los enlaces

Los estados en los que puede estar un enlace son:

* Sin visitar

Aun no se ha pulsado sobre él.

* Sobrepasado

Cuando se pasa el ratón por encima.

* Activo

Acaba de ser pulsado.

* Visitado

Se ha pulsado sobre él con anterioridad.