



UNIVERSIDAD
DE GRANADA

Este documento está protegido por la Ley de Propiedad
Intelectual (Real Decreto Ley 1/1996 de 12 de abril).
Queda expresamente prohibido su uso o distribución
sin autorización del autor.

Tecnologías Web

3º Grado en Ingeniería Informática

Guión de prácticas El lenguaje HTML

1. Objetivo.....	2
2. Aula de prácticas.....	2
Sesión 1.....	3
3. Elementos básicos de HTML.....	4
4. Inferir estructura a partir del diseño.....	5
5. Jerarquía HTML.....	6
6. Enlaces a documentos.....	6
7. Tablas en HTML.....	7
8. Enlaces dentro de una página.....	8
Sesión 2.....	9
9. Creación de formularios.....	10
10. Sitio web personal.....	15
11. Entrega de la práctica.....	17

© Prof. Javier Martínez Baena
Dpto. Ciencias de la Computación e I. A.
Universidad de Granada



Departamento de
Ciencias de la Computación
e Inteligencia Artificial

1. Objetivo

El objetivo de la práctica es desarrollar algunas páginas web sencillas con HTML haciendo especial hincapié en el uso correcto de las etiquetas estructurales del lenguaje. Tenga en cuenta que:

1. No debe preocuparse del aspecto de la página final.
2. Debe centrarse únicamente en redactar código HTML.
3. No puede usar código para dar formato (CSS) **en ningún sitio** salvo, quizá, en aquellos puntos en donde se indique de forma explícita.

Aunque no use elementos de estilo (CSS), es importante que en cada ejercicio tenga un diseño de la página final que resultaría si le aplicase estilo. Esto es importante porque puede que condicione algunos aspectos de la estructura del documento HTML o, al menos, le puede ayudar a definir mejor la estructura del documento.

El diseño previo, aunque puede hacerlo en papel, sería más adecuado hacerlo usando alguna herramienta software que le ayude como, por ejemplo, Pencil.

Esta práctica solo requiere implementar usando el lenguaje HTML y, por lo tanto, no necesita un servidor web. Es decir, podría hacer la implementación en su equipo (local) y visualizar el documento con su navegador. Sin embargo, para probar los formularios, se pide que incluya un pequeño *script* PHP y para esto sí necesita un servidor web funcionando: puede usar `void.ugr.es` o su propio servidor (instalado en su máquina) pero asegúrese de que la página funciona correctamente cuando la suba a `void.ugr.es`.

2. Aula de prácticas

Para el desarrollo de la práctica necesitará:

- Herramientas de desarrollo (editor, etc). Puede usar su ordenador personal o ejecutar una imagen virtual en un PC del aula de prácticas que está preparada para la asignatura.
- Servidor web. El servidor web de la asignatura (`void.ugr.es`) debería estar operativo 24/7 mientras dure el curso académico y puede alojar en él todas sus prácticas manteniendo, en cualquier caso, copias de seguridad en sus dispositivos personales por si este fallase. Puede usar este servidor, su propio ordenador personal si lo tiene preparado o bien el PC del aula de prácticas (la imagen virtual ya incluye un servidor web Apache+MySQL+PHP).

Use la configuración que use, debe tener en cuenta que la corrección de todas las prácticas de la asignatura se realizará sobre el servidor web de la asignatura (`void.ugr.es`) por lo que se recomienda que se familiarice con él.

En caso de usar un PC de la ETSIIT debe usar el siguiente código de arranque:

```
ubutwebvi
```

El usuario y clave para todos los servicios y aplicaciones son:

```
Usuario:    aulas
Clave:      aulas
```

En la imagen cargada dispone de las siguientes herramientas para el desarrollo web:

- Stack LAMP operativo (Apache+PHP+MySQL).
- Editores de texto (sublime, atom, geany, visual studio code). Sublime incluye Emmet.
- Navegadores (Firefox, Chrome, Opera).
- Administración MySQL (MyWebSQL, PHPMyAdmin, MySQL Workbench)
[<http://localhost/mywebsql>] [<http://localhost/phpmyadmin>].
- Cliente SFTP (Filezilla) para subir ficheros a `void.ugr.es` (u otro servidor).
- Koala (preprocesador Sass).
- Prototipado (Pencil).

El servidor Apache está configurado para exportar su directorio `public_html` por lo que su código estará accesible en la URL <http://localhost/~tweb>

Sesión 1

Introducción a HTML

3. Elementos básicos de HTML

Redacte el código HTML de la siguiente página:

Bjarne Stroustrup

Bjarne Stroustrup (n. 30 de diciembre de 1950 en Aarhus, Dinamarca) es un científico de la computación y catedrático de Ciencias de la Computación en la Universidad A&M de Texas. Ha destacado por desarrollar el lenguaje de programación C++. Citando palabras del propio Stroustrup:

Inventé C++, escribiendo sus primeras definiciones, y produciendo la primera implementación... elegí y formulé los criterios de diseño para C++, confeccioné también sus principales utilidades, y fui responsable de elaborar extensas proposiciones en el comité de estándares de C++.

Premios y logros

- 1990. Fue nombrado uno de los "doce mejores científicos jóvenes de América" por la revista Fortune.
- 1995. La revista Byte nombró a Stroustrup una de las veinte personas que más ha influido en la industria informática.
- 2004. Academia Nacional (Estados Unidos) de Ingeniería, 2004.
- 2008. Dr. Dobb's Excellence in Programming award.
- 2019. Doctor Honoris Causa por la Universidad Carlos III.

Libros

1. **Bjarne Stroustrup**, *Programming -- Principles and Practice Using C++*, Addison-Wesley, (diciembre de 2008); ISBN 978-0321543721
2. **Bjarne Stroustrup**, *The C++ Programming Language*, Addison-Wesley Pub Co; Tercera edición (15 de febrero de 2000); ISBN 0-201-70073-5
3. **Bjarne Stroustrup**, *The Design and Evolution of C++*, Addison-Wesley Pub Cp; Primera edición (29 de marzo de 1994); ISBN 0-201-54330-3
4. **Margaret A. Ellis y Bjarne Stroustrup**, *The Annotated C++ Reference Manual*, Addison-Wesley Pub Co; (1 de enero de 1990); ISBN 0-201-51459-1

Puede usar el siguiente texto como base del ejercicio:

Bjarne Stroustrup

Bjarne Stroustrup (n. 30 de diciembre de 1950 en Aarhus, Dinamarca) es un científico de la computación y catedrático de Ciencias de la Computación en la Universidad A&M de Texas. Ha destacado por desarrollar el lenguaje de programación C++. Citando palabras del propio Stroustrup:

Inventé C++, escribiendo sus primeras definiciones, y produciendo la primera implementación... elegí y formulé los criterios de diseño para C++, confeccioné también sus principales utilidades, y fui responsable de elaborar extensas proposiciones en el comité de estándares de C++.

Premios y logros

1990. Fue nombrado uno de los "doce mejores científicos jóvenes de América" por la revista Fortune.

1995. La revista Byte nombró a Stroustrup una de las veinte personas que más ha influido en la industria informática.

2004. Academia Nacional (Estados Unidos) de Ingeniería, 2004.

2008. Dr. Dobb's Excellence in Programming award.

2019. Doctor Honoris Causa por la Universidad Carlos III.

Libros

Bjarne Stroustrup, *Programming -- Principles and Practice Using C++*, Addison-Wesley, (diciembre de 2008); ISBN 978-0321543721

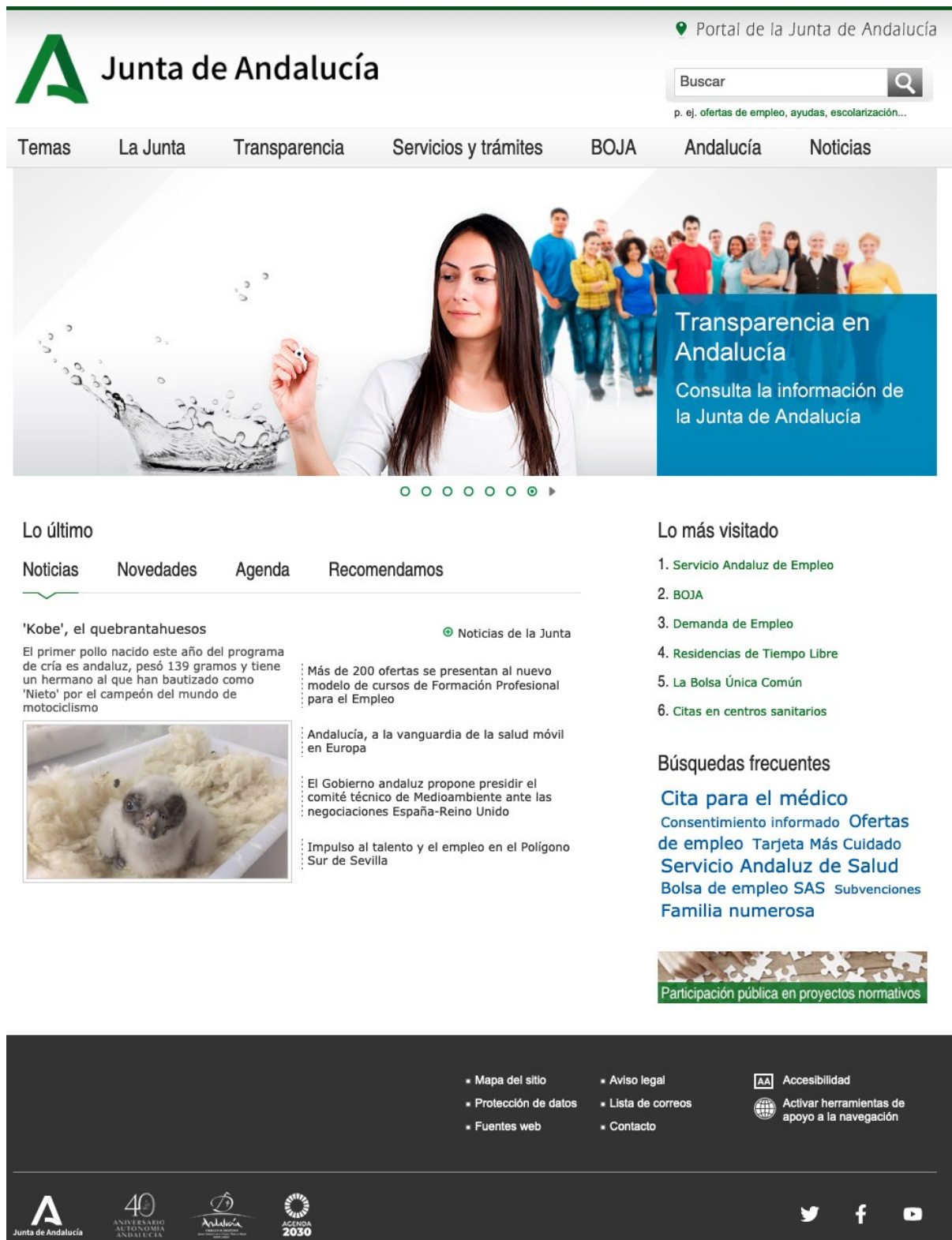
Bjarne Stroustrup, *The C++ Programming Language*, Addison-Wesley Pub Co; Tercera edición (15 de febrero de 2000); ISBN 0-201-70073-5

Bjarne Stroustrup, *The Design and Evolution of C++*, Addison-Wesley Pub Cp; Primera edición (29 de marzo de 1994); ISBN 0-201-54330-3

Margaret A. Ellis y Bjarne Stroustrup, *The Annotated C++ Reference Manual*, Addison-Wesley Pub Co; (1 de enero de 1990); ISBN 0-201-51459-1

4. Inferir estructura a partir del diseño

A continuación puede ver una captura de una página web de la Junta de Andalucía:



A partir de ella, debe escribir el código HTML básico con la estructura del sitio. No tenga en cuenta detalles de aspecto o contenidos concretos. Céntrese en redactar la estructura principal del sitio prestando atención a la jerarquía de los elementos que aparecen y al uso de elementos estructurales de HTML.

5. Jerarquía HTML

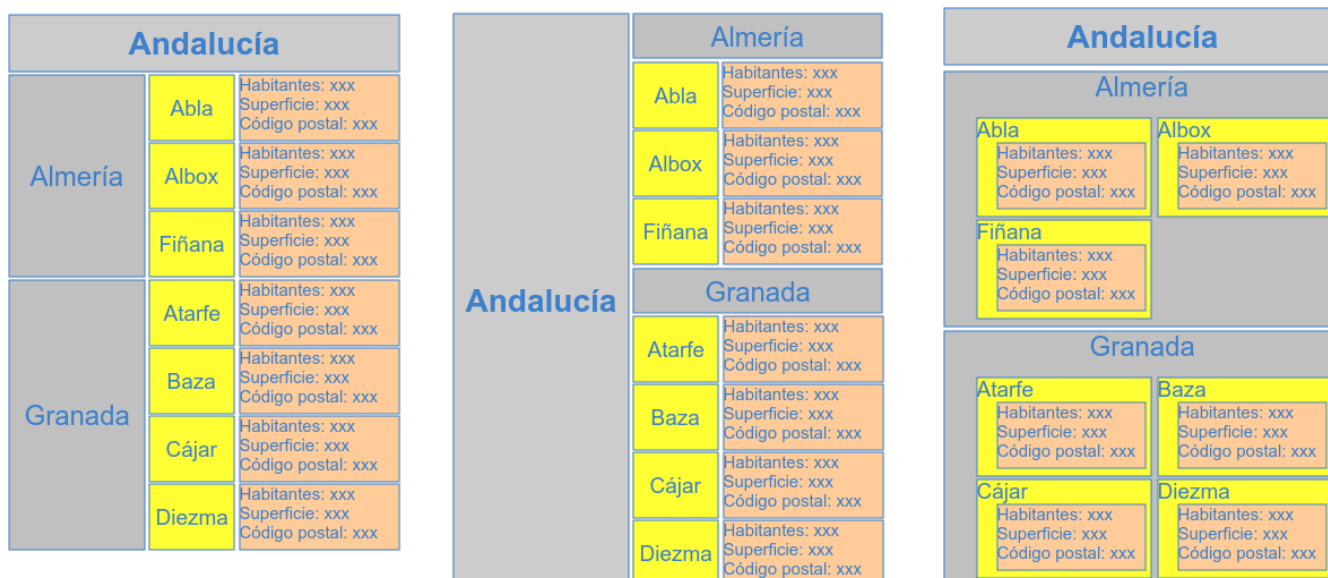
Suponga que está diseñando una web para mostrar información geopolítica sobre poblaciones españolas. Concretamente mantendrá un listado de poblaciones agrupadas por provincias que, a su vez, estarán agrupadas por comunidades autónomas. Para cada población incluirá los siguientes datos: número de habitantes, superficie y código postal.

Para hacer el ejercicio considere, al menos, la inclusión de:

- 2 comunidades autónomas.
- Para cada comunidad autónoma debe incluir 2 provincias.
- Para cada provincia debe incluir 3 municipios.

El objetivo es crear una página HTML en la que esté reflejada la estructura jerárquica de la información indicada. Para ello debería abstraerse de la forma concreta en la que se muestre la información en el navegador y centrarse únicamente en la forma en la que se deberían agrupar, jerárquicamente, los elementos.

En próximos temas se estudiará la forma de maquetar el aspecto visual de los elementos de un documento HTML. A modo de ejemplo, en la siguiente figura se pueden ver diferentes disposiciones visuales de la información. Todas ellas mantienen la misma estructura (HTML).



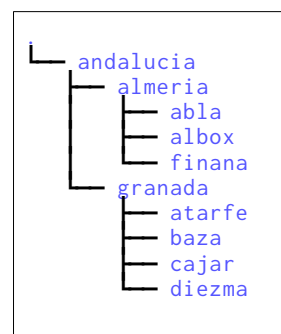
Importante: no debe utilizar elementos de tipo `table` ya que, en ese caso, se estaría definiendo una forma concreta de visualización.

6. Enlaces a documentos

Adapte el ejercicio anterior de manera que la información de cada comunidad, provincia y población esté almacenada en un fichero HTML distinto. El nombre de todos los ficheros deberá ser el mismo: `index.html` y deberá crear una estructura de directorios que refleje la jerarquía comunidad>provincia>municipio tal y como se muestra en la figura.

Es decir, en cada carpeta existirá un documento `index.html` y subcarpetas (esto último solo para comunidades y provincias). El contenido de cada fichero `index.html` será:

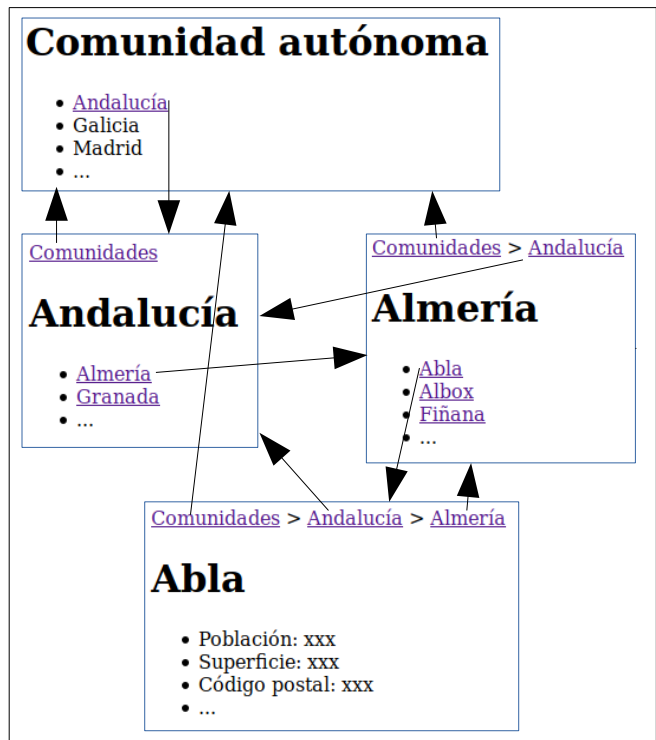
- Al comienzo una cadena de enlaces que permitan al usuario navegar fácilmente dentro de la jerarquía de documentos creada. Este tipo de barras de navegación se conocen como *breadcrumb* o "migas de pan".



- Un título que indica la información contenida en el documento (nombre de la comunidad, provincia, etc).
- Una lista de enlaces a los items que dependen jerárquicamente del documento en cuestión. Cada enlace llevará al documento `index.html` correspondiente.
- En el caso de los ficheros `index.html` relativos a poblaciones se incluirán los datos relativos a la población (número de habitantes, superficie y código postal).

En la figura puede ver el aspecto de algunas de estas páginas a modo de ejemplo. Observe que casi todo con enlaces a otras páginas.

Para realizar el ejercicio considere siempre el uso de URL relativas a la posición de cada documento. De esta forma podría mover todas las páginas a una ubicación diferente de su servidor web y seguiría funcionando correctamente.



7. Tablas en HTML

7.1. Una tabla sencilla

Cree una página web que muestre la tabla de multiplicar de un número (por ejemplo el 4) con el siguiente formato:

M-ando		M-dor		Resultado
4	x	1	=	4
4	x	2	=	8
4	x	3	=	12
4	x	4	=	16
4	x	5	=	20
4	x	6	=	24
4	x	7	=	28
4	x	8	=	32
4	x	9	=	36
4	x	10	=	40

Para que se vean los contornos de las celdas, puede añadir este código CSS en la cabecera del fichero HTML:

```
<head>
...
<style type="text/css">
  td, th { border: 1px solid #CCC; }
  table { border: 1px solid black; }
</style>
</head>
```

Pero recuerde: por ahora no es relevante el aspecto de los elementos que se presentan en la página web.

7.2. Agrupamientos de celdas

Cree una página web que muestre la tabla con la asignación de profesores que se muestra a continuación:

Asignatura			Profesor
Nombre	Grupo	Tipo	
Fundamentos de Programación	A	T	Silvia Acid Carrillo
		P1	David Pelta
		P2	Silvia Acid Carrillo
		P3	Silvia Acid Carrillo
	B	T	Francisco Cortijo Bon
		P1	Francisco Cortijo Bon
		P2	Alberto Fernández Hilario
	C	T	Juan Carlos Cubero Talavera
		P1	Silvia Acid carrillo
P2		Juan Carlos Cubero Talavera	
Metodología de la Programación	A	T	Antonio Garrido Carrillo
		P1	Francisco Cortijo Bon
	B	T	Francisco Cortijo Bon
		P1	Antonio Garrido Carrillo
		P2	Francisco Cortijo Bon

8. Enlaces dentro de una página

Cree una página como la de la figura siguiente. En esta página hay un total de 6 bloques:

- El primero contiene un índice con enlaces a cada una de las secciones que van a continuación.
- Los cinco bloques siguientes contienen, cada uno, un título seguido de un texto y seguido de un enlace al bloque inicial (índice).

<p>Índice</p> <ul style="list-style-type: none"> • Primer párrafo • Segundo párrafo • Tercer párrafo • Cuarto párrafo • Quinto párrafo <p>Primer párrafo</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce consectetur congue tortor in dictum. Sed lectus dolor, pulvinar eu scelerisque in, interdum luctus nibh. Donec nulla odio, pellentesque at leo nec, vestibulum porttitor velit. Nunc diam quam, consectetur sed lacus at, pretium tristique magna. Etiam ut leo laoreet, iaculis est et, eleifend ex. Aenean tincidunt nisi sed est pretium placerat. Nunc interdum nisi mi, eu vestibulum eros consectetur vel. Maecenas vitae aliquet erat. Pellentesque nibh tortor, scelerisque vel dui ac, porta consectetur mi. In sem arcu, pretium at posuere eget, placerat non ligula. Praesent tincidunt nisl justo. Quisque mattis sollicitudin lorem sit amet viverra. Nulla convallis est pretium, facilisis neque at, scelerisque eros. Suspendisse pharetra vulputate ligula eu tincidunt. Nullam sagittis sodales purus sit amet tincidunt. Proin malesuada dui at dolor molestie, in consectetur nibh dignissim.</p> <p>Índice</p> <p>Segundo párrafo</p> <p>Nunc a ligula quis augue gravida posuere vitae pretium nibh. Fusce nec libero aliquet est vestibulum commodo vitae a dolor. Mauris in diam viverra, rutrum leo sit amet, lobortis justo. Sed aliquam mi et nisi volutpat porta. Nam accumsan enim vitae risus auctor iaculis. Donec auctor venenatis aliquet. Aliquam erat volutpat. Nulla tempus venenatis quam quis luctus. Nullam aliquam, leo sit amet tincidunt pharetra, odio massa suscipit ante, non lacinia ipsum elit sed leo. Sed egestas, est in aliquet vulputate, ligula metus bibendum dui, non fermentum elit turpis sit amet purus. In hac habitasse platea dictumst. Vivamus placerat tempus lacus, et pharetra ligula feugiat ac.</p> <p>Índice</p> <p>Tercer párrafo</p>

En <https://www.lipsum.com/> tiene un generador de texto *Lorem Ipsum* para rellenar fácilmente cada bloque.

Sesión 2

Formularios y sitio web completo

9. Creación de formularios

En este ejercicio deberá crear gradualmente un formulario que utilice casi todos los controles estudiados en la asignatura. Para mantener el formulario individual de cada paso se recomienda hacer un único fichero HTML en el que se incluya un nuevo formulario para cada paso (cada nuevo formulario es una copia del anterior con las modificaciones correspondientes).

9.1. Campos de texto

Cree una página web que contenga un formulario con dos campos de texto que simule una página de *login*, es decir, que solicite un nombre de usuario y una clave. El aspecto podría ser como el siguiente:

Datos del usuario
Usuario: Clave:

Para ello debe tener en cuenta las siguientes consideraciones:

- El campo *action* del formulario debe dirigirse hacia un *script* que muestre los datos recibidos.
- La disposición de los elementos (en línea) podría cambiar dependiendo de los *tags* que utilice.
- Asegúrese de usar correctamente los elementos de tipo `label`.
- Debe incluir un botón para enviar los datos.

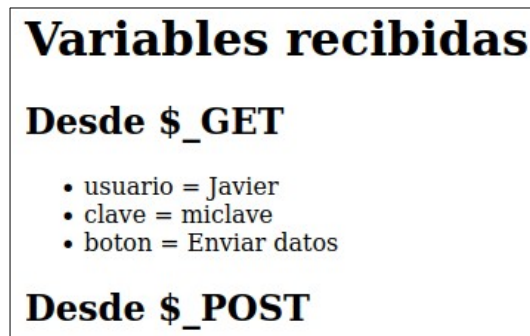
Para comprobar que la información del formulario se envía correctamente, deberá disponer de un *script* PHP en el servidor web que, simplemente, muestre los valores recibidos. A continuación tiene un código a modo de ejemplo que puede usar para este ejercicio. Deberá almacenarlo en un fichero llamado `procesar.php` (observe que debe tener extensión `.php`) e incluir su referencia en el atributo *action* del formulario. A diferencia de los ficheros de tipo `.html` que se pueden ver en cualquier navegador desde el sistema de ficheros local sin necesidad de tener instalado ningún servidor web, los programas PHP deben ser servidos obligatoriamente desde un servidor web que disponga del correspondiente módulo de ejecución de código PHP. Para ello deberá haber realizado correctamente la práctica de instalación del servidor web.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Variables recibidas</title>
  <style type="text/css">
    .valor { color: red; }
  </style>
</head>
<body>
  <?php echo "<h1>Variables recibidas</h1>";
  showVar($_GET, 'Desde $_GET');
  showVar($_POST, 'Desde $_POST');
  function showVar($var, $msg) {
    echo "<h2>$msg</h2>";
    echo "<ul>";
    foreach ($var as $c => $v) {
      if (is_array($v)) {
        echo "<li>$c = <span class='valor'>";
        print_r($v);
        echo "</span></li>";
      } else
```

```
        echo "<li>$c = <span class='valor'>$v</span></li>";
    }
    echo "</ul>";
}
?>
</body>
</html>
```

En caso de que aun no haya configurado un servidor web con PHP, puede usar como URL del atributo action esta: <https://void.ugr.es/procesar.php>.

La siguiente figura muestra un ejemplo de ejecución de este *script* tras pulsar el botón de envío de datos asumiendo que los valores rellenados son usuario "Javier" y clave "miclave":



9.2. Campos obligatorios

Continúe con el formulario y obligue a que el usuario introduzca un valor en el campo "usuario". Para ello debe usar el atributo `required`. Recuerde que esta validación la hace el navegador y que el nivel de soporte puede variar dependiendo del navegador y de su versión.

En ocasiones, aunque algunos campos se hayan etiquetado como obligatorios, es conveniente deshabilitar esas comprobaciones. Pruebe el uso del atributo `novalidate` en la etiqueta `form` para comprobarlo. Esto es particularmente útil durante el proceso de desarrollo para verificar que funciona correctamente la validación en el servidor, que se estudiará más adelante.

9.3. Información al usuario

Incluya un texto informativo en los campos de entrada para advertir al usuario de que el nombre de usuario es obligatorio y de que introduzca una clave en el campo "clave". Utilice el atributo `placeholder`.

Compruebe que, aunque se incluya un texto en `placeholder`, este no se envía al servidor en caso de dejar vacío el campo.

9.4. Valor por defecto

Añada un nuevo campo de texto para indicar la nacionalidad. Dicho campo tendrá como valor por defecto "España".

9.5. Campos especiales de texto

Añada un campo de tipo `email` para introducir el correo electrónico. Compruebe la validación que hace el navegador:

- Si se deja en blanco.
- Si se rellena con algo que no es una dirección de email válida.

¿Influye el navegador/versión en esta validación?

9.6. Formato

En este punto, el aspecto del formulario podría ser similar al que se muestra a continuación (dependiendo de las etiquetas usadas):

Datos del usuario

Usuario:
 Clave:
 Nacionalidad:

Email:

Aunque por ahora el aspecto del formulario no debe preocuparnos, para continuar el ejercicio es conveniente que los campos se muestren de forma un poco más legible. En la siguiente figura se propone que cada campo de entrada se sitúe en una línea independiente. Para ello puede usar varias alternativas:

1. Incluir los elementos `label` e `input`, que son de tipo `inline`, dentro de una etiqueta `p`, que es de tipo `block`.
2. Incluir alguna regla CSS para indicar que determinados elementos de tipo `inline` se comporten como elementos de tipo `block`.
3. Uso de `fieldset` para agrupar conjuntos de campos. Esto se puede combinar con las anteriores.

Datos del usuario

Usuario:

Clave:

Nacionalidad:

Email:

Modifique el formulario para que tenga un aspecto similar al de la figura.

En caso de que para cada entrada utilice un patrón de la forma `<label>Texto: <input ...> </label>` bastaría con incluir el siguiente código CSS dentro del tag `<head>`:

```
<style type="text/css">
  label { display:block; }
</style>
```

9.7. Menú desplegable

Incluya un campo con un menú desplegable para especificar el sexo del usuario al estilo del que se muestra en la figura. Mantenga el mismo orden de la figura e incluya como opción preestablecida la última de ellas.

Compruebe que el valor que se recibe en el servidor coincide con el texto de los items del menú.

Sexo:

Añada un segundo desplegable para que el usuario elija una de las siguientes opciones:

- "Acepta el almacenamiento de mis datos y el envío a terceros".
- "Acepta el almacenamiento de mis datos pero no el envío a terceros".
- "No acepta el almacenamiento ni el envío de datos a terceros".

En casos como este en el que el texto de los items del menú es una cadena más o menos larga y arbitraria y que, incluso, podría modificarse una vez finalizada la aplicación web (piense, por ejemplo, en una web que se muestra en varios idiomas o en un proceso de revisión ortográfica independiente del de desarrollo) puede que no sea buena idea transmitir dicha cadena sino, en su lugar, algún valor que pueda usarse por nuestro código PHP/HTML y que sea transparente al usuario. Utilice el atributo `value` para que el valor enviado en cada una de las opciones sea uno de estos:

- TOTAL (para la primera opción).
- PARCIAL (para la segunda opción).

- NINGUNO (para la tercera opción).

9.8. Opciones excluyentes (*radio buttons*)

Añada un campo para que el usuario indique el idioma preferido para sus comunicaciones mediante *radio buttons* con un aspecto similar al de la figura.

A continuación realice las siguientes tareas:

1. Compruebe qué se envía si no se marca ninguna de las opciones.
2. Modifique el formulario para que esté marcada como opción por defecto una de ellas.

Asegúrese de usar correctamente la etiqueta `label` y verifique que su uso mejora la usabilidad del formulario.

Idioma para comunicaciones:

- ☐ Español
☒ Inglés
☐ Francés

9.9. Opciones múltiples (*checkbox*)

En casos en los que se desea poder marcar cero o más opciones de entre varias se utilizan los campos de tipo *checkbox*. Añada campos de entrada para que el usuario determine sus aficiones en una forma similar a la de la figura.

Ponga especial cuidado en el valor asignado al atributo `name` de la etiqueta `input`. Compruebe qué ocurre si:

1. Utiliza un valor como los que ha usado hasta ahora, por ejemplo `name="aficiones"`.
2. Utiliza un valor diferente de `name` para cada `input`.
3. Utiliza un valor con formato de *array*, por ejemplo `name="aficiones[]"`.

Asegúrese de usar correctamente la etiqueta `label` y verifique que su uso mejora la usabilidad del formulario.

Aficiones:

- ☐ La música
☒ Leer libros
☐ El cine
☐ El deporte

9.10. Soporte de navegadores a campos especiales

Hay varios tipos de campos de entrada introducidos en HTML 5 que no tienen igual soporte en todos los navegadores. Añada campos para introducir:

- Fecha de nacimiento (`date`).
- Cuándo es la semana santa de este año (`week`).

Pruebe a visualizar el formulario en Firefox y en Chrome y visite la web <https://caniuse.com> para comprobar qué soporte tienen.

9.11. Campos ocultos

En ocasiones se necesita que desde el formulario se envíe información adicional a la introducida por el usuario en los distintos controles. Esto se consigue añadiendo un `input` de tipo `hidden`, que se comporta como un campo de texto normal pero que no se muestra.

Incluya un campo oculto que envíe un número de versión del formulario.

9.12. Campos con entrada deshabilitada

Modifique el formulario para que los siguientes campos tengan valores por defecto ya introducidos y que no sean modificables:

- Usuario: pepemartin
- Clave: miclave
- Nacionalidad: España

- Email: pepemartin@miservidor.com
- Sexo: masculino
- Tratamiento de datos: TOTAL
- Idioma para comunicaciones: Inglés
- Aficiones: Leer libros
- Fecha de nacimiento: 01/01/2000
- Cuándo es la semana santa de este año: 18 (semana 18)

Para ello estudie el uso de los atributos `readonly` y `disabled` y compruebe en qué casos se puede o debe usar cada uno y qué efecto tienen sobre el formulario y el envío.

Conteste a esta pregunta: ¿puede tener sentido que un campo que ha sido incluido en el formulario y que se muestra al usuario deshabilitado no se llegue a enviar al servidor?

9.13. Envío de ficheros

Añada un último campo para adjuntar una fotografía del usuario. El aspecto final del formulario será similar a este:

Datos del usuario

Usuario:
 Clave:
 Nacionalidad:
 Email:
 Sexo:
 Tratamiento de datos:
 Idioma para comunicaciones:
☐ Español
☒ Inglés
☐ Francés
 Aficiones:
☐ La música
☒ Leer libros
☐ El cine
☐ El deporte
 Fecha de nacimiento:
 ¿Cuándo es semana santa este año?:
 Fotografía: No se ha seleccionado ningún archivo.

En este formulario, además de incluir el control para la entrada del fichero, debe modificar los atributos de la etiqueta `form`. Además, el *script* PHP que indique en el `action` debe incluir código para procesar el fichero (al menos para informar de que lo ha recibido correctamente). El siguiente código es una modificación del anterior que incluye esa funcionalidad. Puede almacenarlo en un fichero `procesar2.php` y utilizarlo de forma diferenciada en este último formulario:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
```



```

<title>Variables recibidas</title>
<style type="text/css">
    .valor { color: red; }
</style>
</head>
<body>
    <?php echo "<h1>Variables recibidas</h1>";
    showVar($_GET, 'Desde $_GET');
    showVar($_POST, 'Desde $_POST');
    showVarFiles();
    function showVar($var,$msg) {
        echo "<h2>$msg</h2>";
        echo "<ul>";
        foreach ($var as $c => $v) {
            if (is_array($v)) {
                echo "<li>$c = <span class='valor'>";
                print_r($v);
                echo "</span></li>";
            } else
                echo "<li>$c = <span class='valor'>$v</span></li>";
        }
        echo "</ul>";
    }
    function showVarFiles() {
        echo "<h2>Ficheros recibidos:</h2>";
        echo "<ul>";
        foreach ($_FILES as $c => $v) {
            echo "<li>$c = <pre class='valor'>";
            print_r($v);
            echo "</pre>";
            if (is_uploaded_file($v['tmp_name']))
                if (is_image($v['tmp_name']))
                    echo "<img
src='data:image/'.pathinfo($v['name'],PATHINFO_EXTENSION).'.base64,'.base64_encode(file_get_c
ontents($v['tmp_name'])).'>";
                else
                    echo "<p>El fichero no contiene una imagen</p>";
            echo "</li>";
        }
        echo "</ul>";
    }
    function is_image($file) {
        $datos = getimagesize($file);
        return $datos ? ((in_array($datos[2],
        array(IMAGETYPE_GIF, IMAGETYPE_JPEG, IMAGETYPE_PNG, IMAGETYPE_BMP))) ?
        true : false) : false;
    }
    ?>
</body>
</html>

```

Observe que el *script* comprueba si el fichero subido es una imagen y, de ser así, en la respuesta muestra dicha imagen.

10. Sitio web personal

En este ejercicio debe crear el sitio web personal de una persona (preferiblemente usted mismo/a). Dicho sitio constará de las siguientes páginas web:

- Página de entrada

- Página profesional
- Página personal
- Página de contacto

Importante: no haga uso de CSS para hacer el maquetado ni para modificar el aspecto que presentan los diferentes *tags* HTML por defecto. Este ejercicio debe focalizarse en determinar la estructura de la información o, lo que es lo mismo, del documento HTML.

10.1. Estructura del sitio

Lo primero que debe hacer es un *mockup/wireframe/boceto* del sitio, es decir, diseñar la estructura que van a tener cada una de las páginas. Tenga en cuenta que dicha estructura general será la misma para todas ellas, es decir, en todas ellas se repetirán ciertos elementos comunes (título, menú de navegación, pie de página, etc). Para esto puede usar alguna herramienta específica (*Pencil*, etc.) o de uso genérico (*Libreoffice*, etc.).

Realizar ese boceto en este punto tiene varios objetivos:

1. Visualizar la apariencia de la web. Aunque aún no conocemos CSS para hacer el maquetado de la web, nos ayudará a visualizar mejor lo que queremos construir.
2. Estructurar la información. Al construir la estructura de la información lo que estamos haciendo es jerarquizar los contenidos y, por lo tanto, obtener el anidamiento de los diferentes elementos HTML de cada página.
3. Detectar necesidades de la web. En el caso de este ejercicio esto puede no ser evidente por la simplicidad del mismo.

Al ser la estructura general común a todas las páginas, el código HTML de esa parte se repetirá en todas ellas. Por el momento límitese a copiar y pegar dicho código de una página a otra; más adelante resolveremos este problema.

La estructura común deberá incluir los siguientes elementos y usará para los elementos más adecuados de HTML 5:

- Encabezado con el título.
- Menú de navegación con enlaces al resto de páginas.
- Zona principal de contenidos. Los elementos que incluyamos aquí variarán de una página a otra.
- Barra lateral con información secundaria. Puedes incluir algunos datos de interés.
- Pie de página con información típica: copyright, autoría, etc.

10.2. Página principal

Esta será la página que da la bienvenida al sitio web. Al ser la página de entrada para visitar el resto de páginas debe almacenarse en un fichero `index.html`.

Incluirá alguna información contando quién es y deberá incluir, al menos, elementos de tipo encabezado (`h1`, `h2`, ...) y de tipo imagen (`img`).

10.3. Página profesional

En esta página incluirá un currículum vitae con la siguiente información:

- Datos personales (nombre, descripción breve, etc.).
- Datos de contacto (email, teléfono, etc.).
- Resumen profesional. Descripción, en uno o dos párrafos de tu perfil profesional
- Formación académica.
- Historial laboral.

- Competencias / aptitudes / *soft skills*.
- Idiomas.

En esta página:

- Debe hacer uso de elementos de tipo `ol/ul`.
- Debe usar múltiples bloques con `article/section` para las diferentes partes del CV.
- Debe usar algún elemento de tipo `img`.

10.4. Página personal

En esta página describirá sus aficiones y deberá usar al menos un elemento de tipo `table`.

10.5. Página de contacto

Esta página incluirá un formulario de contacto con, al menos, los siguientes campos:

- Nombre.
- Email.
- Texto con la consulta.

Deberá proporcionar el *script* PHP que procesa la información para comprobar el correcto funcionamiento de la página. Puede usar `procesar.php` de ejercicios anteriores.

11. Entrega de la práctica

El profesor dará instrucciones concretas sobre lo que hay que incluir en la entrega de esta práctica así como los plazos para ello. En cualquier caso, el material deberá subirse al servidor web de la asignatura (`void.ugr.es`) y se evaluará únicamente el material subido al servidor.

Para esta práctica en concreto:

- Dentro de su carpeta `public_html` debe incluir una carpeta llamada `"practicaHTML"`.
- Dentro de esa carpeta debe incluir todos los ficheros requeridos.
- Debe existir un fichero llamado `index.html` con la página de inicio del sitio web.
- Debe incluir también los *scripts* PHP para la recepción de formularios.

De esta forma, para acceder al sitio web que ha creado se usará la URL:

`https://void.ugr.es/~USUARIO/practicaHTML`

Observaciones:

- No cumplir con alguno de los requisitos de entrega invalidará la entrega completa.
- No se admitirán entregas por ningún otro medio.

Recomendación: no deje para el último momento la subida de la práctica a `void.ugr.es`. Aunque la práctica es sencilla podría haber algo que no funcione y puede que no tenga tiempo de repararlo.