

DEW – PRÁCTICA 1

PROF. RAMÓN GARCÍA, CURSO 2018/19

DESARROLLO WEB EN EL CLIENTE

Esta práctica a realizar en parejas, con una duración de 3 sesiones, sirve como toma de contacto para el desarrollo de materiales y aplicaciones que tienen como destino el navegador web, y, por tanto, se han de visualizar o ejecutar en el equipo del cliente.

Seguiremos el siguiente guión resumido:

1. Introducción al laboratorio para DEW y a sus herramientas (**apartado 1**)
2. Creación de páginas web (**apartados 2 y 3**)
3. Creación y aplicación de hojas de estilo, incluyendo tipografía y hojas prediseñadas (**apartado 4**)
4. Programación en JavaScript: comprobación de un formulario (**apartado 5**)

El resto de apartados está constituido por información relativa al resultado final esperado y la forma en que se entrega para su evaluación (**apartado 7**), anexos con detalles y código, referenciados en los apartados anteriores.

En cuanto a ordenación temporal, está previsto que los apartados 1, 2 y 3 se cubran en las primeras dos sesiones, y que el apartado 4 se desarrolle en la restante. Se informa sobre la fecha límite y el procedimiento de entrega en el apartado 7.

Lunes (3TI21)	18/02	25/02	04/03
Miércoles (2*3TI1x)	20/02	27/02	06/03

Contenido

1	Introducción y un repaso sobre el laboratorio.....	3
1.1	Interacción mínima con Tomcat: publicación de materiales	4
2	Creación de documentos HTML con Eclipse	6
3	Arreglando documentos.....	8
4	Combinando documentos y estilos	8
4.1	Creando un documento a partir de una referencia	10
5	Validación de formularios en el cliente	14
5.1	Funciones adicionales: calcular letra del NIF	17
5.2	Funciones adicionales: fecha automática	18
5.3	Comprobaciones: letra del NIF.....	18
5.4	Comprobaciones: requisitos de la solución	19
6	Tu servidor virtual y Tomcat.....	21
6.1	Habilitar red, arrancar, detener	22
7	Entrega	23
7.1	Materiales a aportar.....	23
8	Anexo 0. Introducción y preparación de Tomcat.....	24
8.1	Configurando HTTPS en Tomcat (informativo)	25
9	Anexo 1. ej1.html: fuente y visualización (sin estilos)	27
10	Anexo 2. Estilo (azul) y aspecto resultante	32
11	Anexo 3. Tipografía en Google Web Fonts.....	34
11.1	Aspectos destacables (ejemplos con Fundamento):.....	34
12	Anexo 4. dew1.html	36
13	Bibliografía y referencias rápidas	36

1 INTRODUCCIÓN Y UN REPASO SOBRE EL LABORATORIO

Esta práctica ofrece una familiarización con el lenguaje de marcas de hipertexto HTML5 y la forma en que se crean los documentos de este tipo. Como es la primera ocasión en que realizamos esta actividad, será necesario conocer previamente las herramientas con las que contaremos, y cómo pueden servirnos para este cometido.

Se desea recordar que nuestro objetivo final, en la asignatura, no es crear páginas HTML artesanalmente, sino por programa; sin embargo esto no evita que debamos conocer el lenguaje de marcas; más bien al contrario: no podemos usar herramientas interactivas de alto nivel para producir los documentos, y deberemos realizar esta actividad recurriendo directamente al código.

Como a lo largo de la asignatura emplearemos el entorno de desarrollo integrado **Eclipse**, ya conocido, parece razonable **no utilizar otro** para la edición de hipertexto. La distribución JEE de Eclipse ofrece la posibilidad de crear páginas de hipertexto de 2 formas: mediante el editor estándar de resalte sintáctico por colores, o con una capacidad especial que incluye previsualización de la página resultante. Por tanto, salvo por el sobredimensionamiento del entorno para nuestro caso, Eclipse es una herramienta suficiente.

En cualquier caso nos bastaría cualquier editor con resalte sintáctico, como Visual Studio Code¹, y Bluefish, entre otros.

La primera parte de la práctica nos permite familiarizarnos con el proceso de la creación de un documento de hipertexto en Eclipse. Nuestros documentos serán inicialmente sencillos, de forma que nuestro objetivo se centre en el proceso seguido para su creación y publicación básicas. En la segunda incrementaremos la complejidad de los documentos, llegando a proponer la creación de una página similar a otra proporcionada como referencia, interviniendo tipografía y hojas de estilo. Por último, en el apartado 5, se pretende añadir a un formulario algo de código en el cliente que compruebe si los datos rellenados son correctos antes de enviarlos.

Los laboratorios del DSIC permiten la realización de prácticas con diferentes configuraciones en cuanto a aplicaciones y sistemas operativos. Por ello pueden convivir recursos de varias asignaturas que no siempre son adecuados. En relación con DEW, las aplicaciones y aspectos que nos interesan se mencionan a continuación:

- Las prácticas se realizan en LINUX, que será el sistema elegido entre las posibilidades de arranque de los equipos de escritorio del laboratorio.
 - Para este curso 2018/19 la instalación disponible es CentOS 7² de 64 bits, **virtualizada**.
- Se dispone de dos instalaciones diferenciadas:
 1. **Equipo de escritorio**: son los de laboratorio, equivalentes a una sesión en linuxdesktop.dsic.upv.es. Disponen de software para el desarrollo, incluyendo el entorno Eclipse 4.7 Oxygen J2EE, editores de texto, navegadores web, etc.

¹ Instalado en el laboratorio

² Se trata de una distribución de código libre equivalente a RedHat 7

2. Equipo de explotación o **servidor virtual (dew-tulogin-1819.dsic.cloud)**. Alojados en el portal del DSIC, con acceso mediante VPN, incluyen software servidor como el contenedor Tomcat 8.5 de APACHE.

La forma de trabajo simplificada consiste en producir en el escritorio, enviar al servidor y comprobar el funcionamiento accediendo con el navegador del escritorio.

Cada alumno se responsabiliza del servidor virtual que le corresponda. La contraseña inicial para el usuario **root** es **inIz1a1**. Debe ser modificada y recordada en su primer uso.

- Cada alumno puede acceder desde su escritorio a una zona de materiales de cada asignatura, con una ruta que comienza en el directorio de nuestra cuenta (\$HOME/asigDSIC/ETSINF/dew)

1.1 Interacción mínima con Tomcat: publicación de materiales

Se denomina “**publicación**” a la acción de dar acceso al documento o aplicación web que nos interese. En nuestro caso equivale a copiar nuestro desarrollo a un lugar concreto del servidor virtual junto con alguna acción para solicitar su despliegue.

Tenéis un resumen de la operativa con vuestras virtuales de portal en el *Anexo. Uso de Servidores Virtuales* del boletín de la práctica 1 de TSR. En la zona de recursos de DEW en PoliformaT encontrareis una copia en tres idiomas de ese anexo.

Nuestro caso, en esta primera práctica, es extremadamente simple porque únicamente desarrollamos *materiales*. Comencemos con una carpeta dew1, en la que incluimos el archivo dew1.html del anexo 4. La publicación y verificación consistirá en:

1. **En nuestro escritorio:** Comprimir el material a publicar (la carpeta dew1). Obtenemos dew1.tgz

```
tar czvf dew1.tgz dew1
```

2. **En nuestro escritorio:** Copiar el archivo comprimido a la carpeta /root/ de la virtual

```
scp dew1.tgz root@dew-tulogin-1819.dsic.cloud:/root/tomcat/webapps
```

3. **Desde nuestro escritorio:** Iniciamos sesión en nuestro servidor

```
ssh root@dew-tulogin-1819.dsic.cloud
```

4. **En nuestro servidor:** Extraemos el contenido del archivo pasándolo al directorio webapps/dew1 de Tomcat.

```
cd /root/tomcat/webapps
tar xzvf dew1.tgz
```

5. **En nuestro servidor:** Iniciamos (si se encontraba detenido) el contenedor Tomcat con el script tomcat/bin/startup.sh

```
/root/tomcat/bin/startup.sh
```

6. **En nuestro escritorio:** Comprobamos mediante un navegador el resultado mediante el URL `http://dew-tulogin-1819.dsic.cloud:8080/dew1/dew1.html`

Las modificaciones necesarias se realizan localmente, se publican y se vuelven a comprobar con el navegador siguiendo el ciclo descrito.

7. **En nuestro servidor:** Al terminar la sesión detenemos el contenedor Tomcat con el script `/root/tomcat/bin/shutdown.sh`

Recuerda que, para impedir el plagio, no debes dejar accesible tu material a otr@s compañer@s. Es tu responsabilidad.

Los recursos que se ponen a disposición del alumno se justifican por su carácter educativo, adecuado para las competencias que han de adquirir. Los materiales almacenados en entornos compartidos requieren una cautela especial cuando de ellos depende una calificación que puede estar condicionada por la duda sobre la autenticidad de la autoría.

Merece la pena **destacar que el uso de materiales y soluciones procedentes de cursos anteriores está especialmente contemplado**, aunque se tratase de *autoplagio*.

- El uso de dichos materiales y/o soluciones es un fraude que se califica con un 0.

No dejes tu máquina virtual en funcionamiento mientras no la utilices. Tienes más detalles sobre la interacción con tu servidor virtual y Tomcat en el apartado 6.

2 CREACIÓN DE DOCUMENTOS HTML CON ECLIPSE

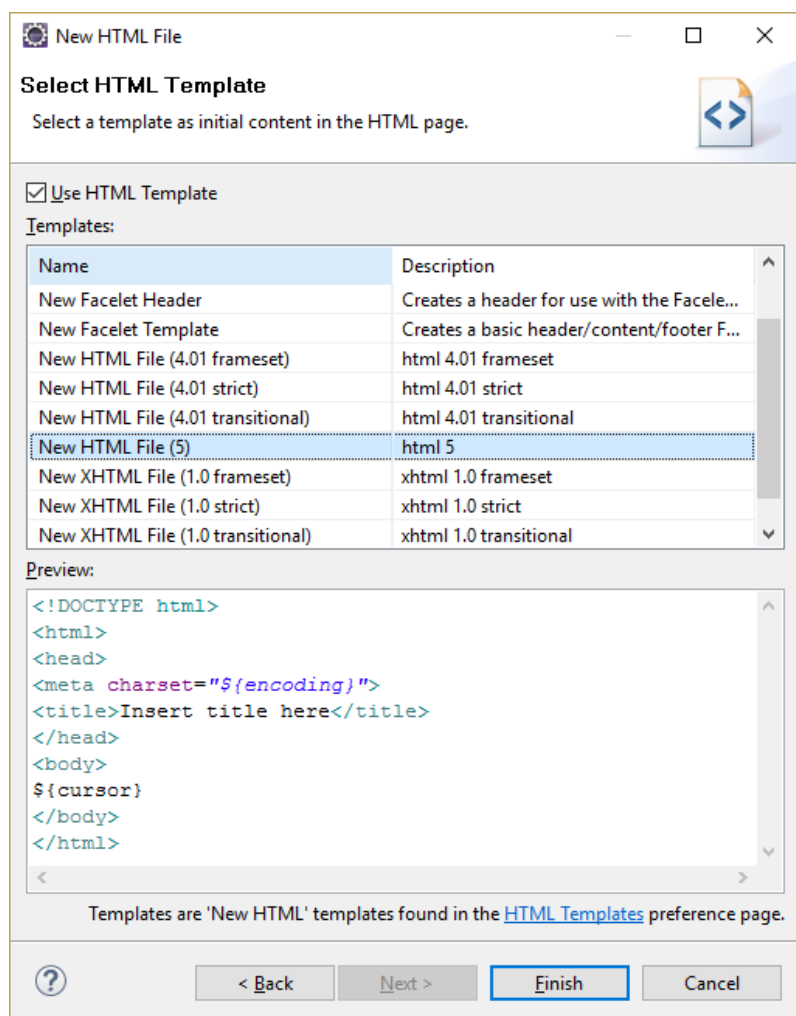
El entorno de desarrollo integrado Eclipse JEE permite crear dos tipos de proyecto para la web:

1. Los **estáticos** no incluyen programas (servlets) que deban ejecutarse en el servidor, limitándose por tanto a documentos HTML, CSS y JavaScript.
2. Los **dinámicos**, que sí permiten crear aplicaciones ejecutables en el servidor, denominadas servlets cuando se diseñan para ejecutarse en un contenedor Java. En este caso se requerirán las clases que acompañan al contenedor.

En este momento no necesitamos construir aplicaciones del segundo tipo, de modo que los proyectos que elijamos crear tendrán carácter estático. Toda la práctica 1 será parte del proyecto web estático **dew1**.

Ejecutaremos Eclipse, encuentra disponible en el menú Desarrollo, para crear nuestro proyecto **dew1** y una página HTML (**p1.html**) dentro del mismo. La parte novedosa aparece cuando el asistente entra en marcha para proponernos la selección de una plantilla HTML; esto es, podemos elegir una estructura de la página con una cabecera compatible con alguna de las versiones de HTML, a elegir.

- Seleccionamos la plantilla **HTML 5**.
- Puede observarse en la ilustración que el archivo a editar contendrá ya las órdenes HTML preliminares imprescindibles para esta página.



Escribiremos el documento que se muestra en la ilustración de la derecha, con dos alternativas:

1. En primer lugar introducimos el título y los dos primeros párrafos en el editor “convencional”. Observamos que colorea el texto dependiendo de la sintaxis.
2. En segundo lugar, cerramos la ventana del editor y pulsamos sobre el archivo con el botón derecho, para que aparezca el menú contextual, en el que seleccionaremos Open With -> Web Page Editor

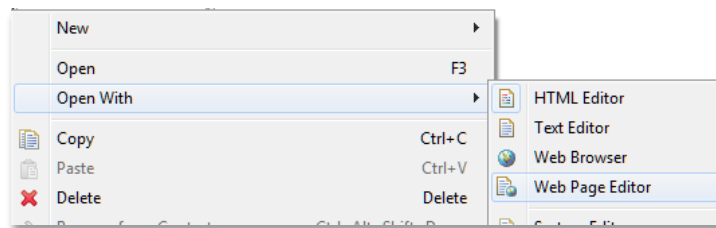
El título

Un ejemplo muy sencillo

Una lista

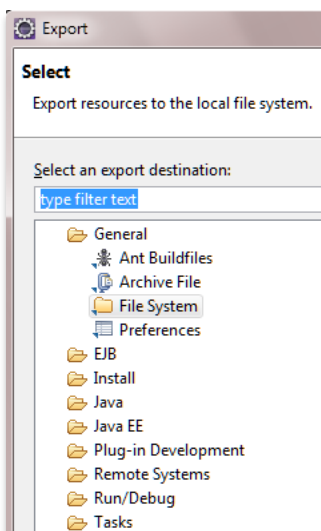
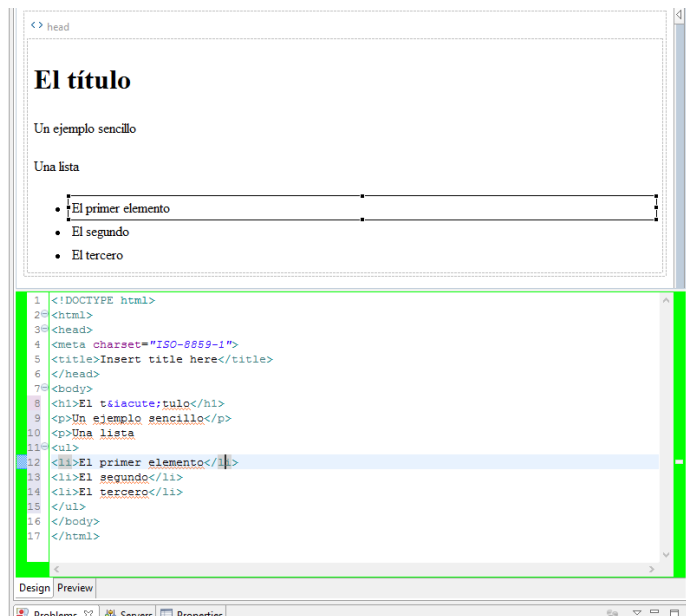
- El primer elemento
- El segundo
- El tercero

Se acabó



Este editor divide la ventana en dos partes horizontales, comportándose la inferior como el editor que conocemos, y la superior como editor WYSIWYG³ o previsualizador

Los errores sintácticos suelen ser señalados sobre el texto, y diagnosticados por el editor como se muestra en la ilustración de la derecha.



Cuando ya no queden errores que resolver, exportaremos la página resultante a un archivo.

³ What You See Is What You Get

3 ARREGLANDO DOCUMENTOS

Ya sabemos que los navegadores son muy tolerantes con los documentos que incumplen la especificación estándar de HTML que ellos mismos anuncian. Por ello el siguiente ejercicio pretende que encontréis y arregléis (para que se ajuste a HTML 5) los defectos de un documento (**p2MAL.html**, que referencia a **lucene_demo.png**).

- Lo podéis encontrar en el directorio `$HOME/asigDSIC/ETSINF/dew/labo1` del laboratorio
- Está pensado para HTML 4, pero deberéis indicar en la cabecera que pretendéis usar HTML 5 y arreglarlo para que realmente cumpla con esa especificación.

Una vez finalizado sin problemas, su nombre debe ser **p2.html**. Lee primero el anexo 4 que te informa sobre cómo organizar los materiales entregables en esta práctica.

- En <https://html.spec.whatwg.org/multipage/obsolete.html> puedes conocer los elementos descartados o reemplazados en HTML 5. También, en general, https://wiki.whatwg.org/wiki/Presentational_elements_and_attributes te ayuda a diferenciar entre marcas de aspecto y de semántica.

Es *imprescindible* someter el resultado a un validador, como <https://validator.w3.org/>, asegurándose que no presenta errores respecto a HTML 5.

- Si el servicio indicado no se encontrara disponible, podéis emplear el validador local de `http://memex.dsic.upv.es:8080/vnu/`

En ocasiones puede haber una discrepancia entre el formato de la codificación empleado (depende del editor y del Sistema Operativo) y el especificado en el documento (en nuestro caso siempre indicamos `utf-8`).

- En LINUX podemos disponer de la orden `iconv` para realizar la conversión:

```
iconv -f ascii -t utf-8 fichero_original -o nuevo_fichero
```

4 COMBINANDO DOCUMENTOS Y ESTILOS

A la hora de realizar una aplicación web suele otorgarse gran importancia a su componente visual. Los diseños elegantes y con alto impacto visual permiten captar la atención de un mayor número de gente, y por tanto pueden ser un factor clave en el éxito de la aplicación.

Los objetivos de esta asignatura no incluyen la capacitación en el ámbito estético, por lo que nuestros propósitos en esta área serán modestos.

Desde hace años en las aplicaciones web se ha implantado la separación entre el contenido (fichero HTML) y el diseño (fichero CSS, por *Cascade Style Sheet*), de forma que las hojas de estilo (CSS) se han convertido en un mecanismo básico para trabajar en el aspecto visual, permitiendo modificarlo cuanto sea necesario sin necesidad de implicar cambios en el contenido (HTML). Se pueden encontrar aplicaciones que permiten editar tanto las CSS como el código HTML de una página cargada, y comprobar *in situ* el resultado obtenido al realizar dichos cambios. Algunas muy útiles pero sencillas son:

- X-Ray Googles, de Mozilla, para inspeccionar código: <https://goggles.mozilla.org/>
- Thimble, también de Mozilla, para aprender HTML y CSS: <https://thimble.mozilla.org/es/>
- El complemento *Web Developer* para Firefox: <https://addons.mozilla.org/en-US/firefox/addon/web-developer/>

Con este último complemento observaremos que aparece una barra extra con varias utilidades de interés para un diseñador web, destacando:

- CSS -> Edit CSS
- Miscellaneous -> Edit HTML
- Tools -> Validate HTML

Nuestras actividades relacionadas con el “diseño web” nos permitirán aprender que:

- Una página, con el mismo contenido, puede cambiar de aspecto dependiendo de la hoja de estilo que se utilice.
- Pueden usarse recursos tipográficos gratuitos elaborados por otros (Google Web Fonts)
- Pueden usarse hojas de estilo⁴ realizadas por otros (Bootstrap)

El objetivo de familiarizarnos será suficiente al principio, pero más tarde deberemos aplicar lo aprendido para modificar el documento de forma que **aproveche** la funcionalidad de Bootstrap; es decir, deberemos llegar a entenderlo para aprovechar sus funciones en una página.

La página⁵, sin hoja de estilo tiene como URL <http://memex.dsic.upv.es:8080/dew1/ej1.html>⁶

Podéis observar tanto su aspecto como su contenido. Podéis probar con los formularios, que invocan un servlet con URL <http://memex.dsic.upv.es:8080/dew1/Printenv>

Es interesante leer el código fuente de la página, observando que no hay mención a ningún estilo en la cabecera, pero sí que encontramos varios párrafos en las primeras líneas que poseen un atributo `class="nota"`.

- Esto puede ser aprovechado en algún estilo que establezca propiedades específicas para estos párrafos.

El documento en sí puede tener otro interés como ilustración de varias construcciones HTML, tanto cabeceras, listas, tablas y elementos de formulario.

Sigue los pasos del **anexo 1** para detalles.

⁴ ¡y muchas más cosas!

⁵ Su código fuente aparece como anexo 1

⁶ También disponible en PoliformaT

En una segunda parte, el efecto de aplicar una hoja de estilos concreta se puede observar en <http://memex.dsic.upv.es:8080/dew1/ej1Montserrat.html> y <http://memex.dsic.upv.es:8080/dew1/ej1Fondamento.html>

- El primero carga la tipografía de <http://fonts.googleapis.com/css?family=Montserrat> y una hoja de estilo que utiliza el tipo de letra 'Montserrat'.
 - Debes averiguar el URL y descargar la hoja de estilo. Anota la orden que afecta a esos párrafos con atributo `class="nota"`
- El segundo, carga la tipografía de <http://fonts.googleapis.com/css?family=Fondamento> y una hoja de estilo que utiliza ese tipo de letra.
- El código fuente de ambos ha sido *retocado* para dificultar el plagio. No es buena idea intentar reconstruir el original a partir de la versión retocada: con el mismo esfuerzo se puede aprender a resolverlo de la forma apropiada.

Sigue los **anexos 2 y 3** para detalles acerca de la actividad a realizar.

4.1 Creando un documento a partir de una referencia

Este apartado responde al supuesto en que deseamos construir una página similar a un diseño conocido. Como el perfil de esta asignatura no cumple con las necesidades del diseño gráfico, en lugar de construir nuestra propia hoja de estilo usaremos un sistema de amplia aceptación: **Bootstrap** (www.getbootstrap.com), que incluye hojas de estilos junto con otras funcionalidades, pero que todavía necesitan un paso para concretar su aspecto.

Complementando a Bootstrap encontramos **Bootswatch** (bootswatch.com), que incluye una serie de personalizaciones agrupadas como temas para Bootstrap.

En Bootswatch podemos seleccionar los temas que nos interesen (en nuestro caso son *Superhero* y *Sketchy*); para cada uno de ellos se proporcionan 2 archivos CSS (`bootstrap.css` y `bootstrap.min.css`) que debemos colocar en el directorio de Bootstrap para personalizarlo.

Por tanto, para su uso:

1. Descargamos y descomprimos Bootstrap 4, de manera que podamos referenciar sus hojas de estilo desde nuestras páginas.
2. Hacemos tantas copias del directorio resultante como pruebas deseemos efectuar (en nuestro caso al menos 2).
3. Desde Bootswatch descargamos los archivos que han de sustituir a sus correspondientes en Bootstrap (directorios CSS). Hay instrucciones para usarlo.

El objetivo final consiste en construir una página con cada uno de estos dos estilos, como puedes ver⁷ en <http://memex.dsic.upv.es:8080/dew1/superhero/ej1.html> y en <http://memex.dsic.upv.es:8080/dew1/sketchy/ej1.html>

⁷ Su contenido ha sido *ofuscado*

DEW1718/Superhero

Inicio Los títulos ... Área de texto Pie

Toda página debería tener un título.

Claro que después habrá que poner el texto normal.

Recuerda que siempre puedes ver el código fuente.

Tampoco deberías olvidar que se pueden guardar cambios necesarios.

Cuando uses Eclipse, el editor de páginas web te proporciona su previsualización.

Cuando uses un editor de texto llano, te aconsejo que uses el editor elegido. De esta manera podrás observar los cambios en la página.

Cuidado con los derechos de autor si deseas emplear imágenes de Commons (CC BY) dispones de un catálogo de imágenes.

Los títulos de nivel inferior destacan aspectos relativamente menos importantes.

Los de nivel 3 destacan todavía menos.

Y no deberías pasar de los de nivel 4.

Observarás que no destacan demasiado respecto al texto llano.

Es bueno practicar con las listas.

La primera será una lista no ordenada para enumerar:

DEW1718/Sketchy

Inicio

Los títulos ...

Área de texto

Pie

1. Batir los huevos.
2. Añadir el tomate.
3. Salpimentar al gusto.
4. Freír a fuego fuerte.
5. Retirar cuando comience a dorarse.
6. Servir a temperatura ambiente.

Hay otros tipos de lista que son más complicadas, pero su utilidad estriba en:

Conveniencia

Puesto que hay momentos en los que nos interesa separar el término a definir de su definición.

Claridad

Dado que esta separación destaca y separa el término a definir.

Versatilidad

Podríamos construir un programa que calculase un índice de términos seleccionando los que se han definido.

Desde luego que tenemos la posibilidad de elegir los bullets (también denominados viñetas en algunos programas) o el sistema de numeración a aplicar (p.ej. alfabético), pero esas capacidades añadidas son un extra en este momento. Recuerda que se obtienen mediante atributos aplicados a las listas.

No sólo de listas vive el hombre ... también se necesita tener tablas

Es difícil encontrar argumentos nuevos para defender la existencia de esta estructura. Un ejemplo sencillo podría ser un calendario de la semana laboral:

Lunes	Martes	Miércoles	Jueves	Viernes
26	27	28	01	02

Pero quizá esto fuera demasiado obvio.

¡- y esto ha sido todo por hoy!

¡y te recuerdo:

Sólo podrás elegir una de las siguientes opciones (la segunda tiene ya una marca por omisión):

- ☐ Primera opción.
☒ Segunda opción.
☐ Tercera opción.

Segunda prueba

Comencemos el tercer formulario. Aquí usaremos listas y menús desplegables.

Dime de dónde vienes:

En ocasiones uno puede elegir varios valores si no son incompatibles. Hace falta pulsar la tecla de mayúsculas junto con el botón del ratón si se desea marcar varias opciones.

Dime por qué estudias Informática:

Tercer formulario

Una forma organizada de resolver este ejercicio consiste en centrarnos en cada ocasión en uno de los aspectos: primero nos preocupamos de la estructura de la página, y después de la decoración de la misma.

1. Sobre la estructura:

Para poder aplicar Bootstrap a nuestras páginas debemos tener presentes las siguientes consideraciones:

- Cualquier página realizada con Bootstrap nos puede servir de ejemplo de uso (ver <http://getbootstrap.com/getting-started/#examples>). Para ello, basta con cargarla en el navegador y leer su código HTML.
- Podemos seleccionar uno de los ejemplos de Bootstrap **que se asemeje a la estructura que deseamos en nuestra página**: barra de navegación, zona destacada, varias columnas y pie de página final. Un candidato es *Jumbotron* (o alguna variante). Debemos estudiar su código fuente para entender la estructura de la página, y localizar en qué lugares de la misma se especifica la barra de navegación, etc.
- A continuación, seleccionaremos los fragmentos relevantes de `ej1.html` que deban aparecer en cada lugar, consiguiendo que el contenido se adapte a esa estructura. No modificamos la hoja de estilo todavía.
- El ejemplo seleccionado de Bootstrap actúa como esqueleto para organizar nuestro contenido.
- En la parte inferior de toda la página se muestra una despedida ("*¡... y esto ha sido todo por hoy!*"), que debe colocarse al final del documento (¡no de la primera columna!)

Obsérvese que esta lista de acciones no es completa sino orientativa. Una vez resuelta la parte de la estructura, conviene disponer de dos copias sobre las que realizar las dos versiones.

2. Sobre el aspecto, debes considerar:

- La visualización depende del navegador y otros factores, y hay un margen de variabilidad admisible.
- Obsérvese los **márgenes** laterales, no como un problema referente a la medida exacta a incluir, sino como un **síntoma del anidamiento correcto** o incorrecto de las piezas que constituyen la página.
- La barra superior ha de permanecer fija cuando se realiza un desplazamiento (*scroll*) vertical de la página. Está ideada como una barra de navegación siempre visible.
- En dicha barra hay entradas sirven para la navegación local dentro de la página.
 1. **Inicio** debería conducirte al comienzo de la página
 2. **Los títulos...** deberían desplazar el contenido hasta la sección que comienza con "*Los títulos de nivel inferior ...*"
 3. **Área de texto** haría algo similar referente al lugar con el `textarea`
 4. **Pie** produciría un desplazamiento vertical para llegar hasta la última línea de la página.
- No es necesario resolver el defecto del solapamiento de la barra de navegación sobre los contenidos de la página (oculta parte que deseamos ver).

- Juega con las dimensiones de la ventana del navegador, observando que el texto y las cajas se ajustan dinámicamente a esas variaciones.

La fuente de información más importante son las propias páginas de Bootstrap, pero siendo un sistema tan popular podréis encontrar información en otros muchos lugares.

5 VALIDACIÓN DE FORMULARIOS EN EL CLIENTE

En este apartado hacemos frente a un problema conocido: comprobar que los valores introducidos por el usuario se adaptan a ciertas restricciones, propias del tipo de información que se desea recibir.

Al realizar esta validación dentro del propio navegador, conseguimos mandar la información al servidor únicamente cuando los datos introducidos sean considerados correctos. Como consecuencia, en caso negativo, se le ofrecerá inmediatamente al usuario la posibilidad de enmendar los errores junto con un diagnóstico explicativo.

- De todos modos la validación en el navegador **no nos libera de efectuar las comprobaciones pertinentes** en la parte servidora.

El formulario se puede observar en la figura de la derecha.

El detalle de los requisitos aparece en el apartado 5.4.

Formulario a comprobar

Datos del producto

Código

Precio por unidad
Proporciona un valor numérico

Datos del cliente

DNI
8 dígitos, sin la letra

Apellidos, Nombre
Separados por una coma

Correo electrónico
No olvides la @

Dirección de envío

Departamentos visitados
Selecciona al menos uno

Alimentación
Muebles
Música

Datos de la tarjeta de crédito

Tipo de tarjeta
Selecciona el emisor

☐ Visa
☐ MasterCard
☐ American Express

Número de tarjeta
12 dígitos

Repite el número
12 dígitos

Enviar Orden

El código fuente de la página (**formToCheckDEW.html**) con el formulario que se ofrece como punto de partida es:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <title>Formulario a comprobar</title>
  <link rel="stylesheet" href="tsr.css" />
</head>
<body>

  <div id="tsrStyle" class="tsrForm">
    <h1>Formulario a comprobar</h1>
    <form action="http://memex.dsic.upv.es:8080/dew1/Printenv" id="clientForm"
method="post">
      <fieldset><legend>Datos del producto</legend>

        <label for="codigo">Código</label>
        </label> <input class="textbox" name="codigo" id="codigo" type="text" />
        <label for="precio">Precio por unidad <span class="small">Proporciona un
valor numérico</span>
        </label> <input class="textbox" name="precio" id="precio" type="text" />

      </fieldset>
      <fieldset><legend>Datos del cliente</legend>

        <label for="dni">DNI<span class="small">8 dígitos, sin la letra</span>
        </label> <input class="textbox" name="dni" id="dni" type="text" />
        <label for="nombre">Apellidos, Nombre<span class="small">Separados por una
coma</span>
        </label> <input class="textbox" name="nombre" id="nombre" type="text" />
        <label for="email">Correo electrónico<span class="small">No olvides la
@</span>
        </label> <input class="textbox" name="email" id="email" type="text" />
        <label for="direccion">Dirección de envío
        </label> <textarea class="textbox" name="direccion" id="direccion" rows="3"
cols="40"></textarea>

        <label for="select">Departamentos visitados <span class="small">Selecciona al
menos uno</span>
        </label> <select class="textbox" name="select" id="select" size="3"
multiple="multiple">
          <option value="Alimentacion">Alimentación</option>
          <option value="Muebles">Muebles</option>
          <option value="Musica">Música</option>
          <option value="Otros">Otros</option>
        </select>

      </fieldset>
      <fieldset><legend>Datos de la tarjeta de crédito</legend>
        <label for="tarj1">Tipo de tarjeta<span class="small">Selecciona el
emisor</span>
        </label>

        <fieldset class="radio">
          <ul>
```

```

        <li><input class="radiobutton" name="tarjeta" id="tarj1" value="Visa"
type="radio" />Visa</li>
        <li><input class="radiobutton" name="tarjeta" id="tarj2"
value="MasterCard" type="radio" />MasterCard</li>
        <li><input class="radiobutton" name="tarjeta" id="tarj3" value="Amex"
type="radio" />American Express</li>
    </ul>
</fieldset>

    <label for="numTarj1">Número de tarjeta<span class="small">12
dígitos</span>
    </label> <input class="textbox" name="numTarj1" id="numTarj1"
type="password">

    <label for="numTarj2">Repite el número<span class="small">12
dígitos</span>
    </label> <input class="textbox" name="numTarj2" id="numTarj2"
type="password">

</fieldset>
<button type="submit">Enviar Orden</button>

</form>
</div>
</body>
</html>

```

Cumplimentando los datos solicitados y pulsando sobre el botón *Enviar Orden*, se recibirá como respuesta la información ofrecida por el servlet `Printenv`ⁱ de `memex.dsic.upv.es` (indicado como valor del atributo `action` del formulario).

- Puedes observar dicho resultado en la ilustración de la derecha.

Las modificaciones que necesitamos aplicar a este formulario incluyen añadir funciones que inserten información (letra del NIF y fecha automática), y comprobar la validez de los valores aportados por el usuario. Algunas de estas modificaciones se comentan en detalle mientras que en otras solo indicamos qué se pretende conseguir.

Para ilustrar estas funciones y comprobaciones, nos basaremos en el cálculo y verificación de la letra del NIF.

Información accesible desde este servlet

Variables en el URL

VARIABLE	VALOR
codigo	patatas12
precio	334,44
dni	77232917
nombre	Nito del Bosque, Elena
email	elena@dot.com
direccion	c/Bajo LaSeta, 1 59116 Yellowstone USA
select	Alimentacion
select	Musica
tarjeta	Amex
numTarj1	214748364700
numTarj2	214748364700

5.1 Funciones adicionales: calcular letra del NIF

El propósito consiste en procesar el valor de la variable de formulario `dni`, cuando se pulse el botón de envío, para añadirle la letra del NIF correspondiente.

- Las situaciones de error se estudiarán en un apartado posterior sobre comprobaciones.

La fórmula *oficial* para ese cálculo se puede consultar en el ministerio de Interior <http://www.interior.gob.es/web/servicios-al-ciudadano/dni/calculo-del-digito-de-control-del-nif-nie>. Brevemente, si el DNI es `n`, la letra `l` correspondiente se calcula como:

```
letras="TRWAGMYFPDXBNJZSQVHLCKET";
l = letras[n % 23];
```

- ¿Cómo obtenemos `n`? En nuestro caso se trata de `clientForm.dni.value`.
- ¿Qué hacemos con el valor `l` obtenido? Agregarlo (concatenarlo) al final del `dni`.
- ¿Cómo visualizamos el resultado? En el propio formulario, sustituyendo el valor introducido por el usuario.
- ¿Cuándo se lleva a cabo la operación? Al pulsar el botón de envío. Interceptamos el evento `submit` del formulario para que ejecute una función JavaScript en la que modificaremos el `dni`.
 - En un primer intento, cancelamos el envío (observamos el resultado en el formulario).
 - En el segundo, completamos el envío (observamos el resultado en la respuesta de `Printenv`).

Llamemos `calculaLetraNIF` a la función. El propio formulario se coloca como parámetro en la intercepción del evento `submit`:

```
<form action="http://memex.dsic.upv.es:8080/dew1/Printenv" id="clientForm"
method="post" onsubmit="return calculaLetraNIF(this);">
```

Y en la cabecera del documento HTML escribimos el código de `calculaLetraNIF`:

```
<script language="javascript" type="text/javascript">
function calculaLetraNIF(f)
{
    letras="TRWAGMYFPDXBNJZSQVHLCKET";
    f.dni.value=f.dni.value+"-"+letras[f.dni.value % 23];
    return false; // impide que se invoque el CGI del campo action
}
</script>
```

De esta manera nos podremos centrar en el funcionamiento de la invocación y en el acceso a las variables del formulario.

Si supera esta prueba, sustituimos `false` por `true` en la función, de manera que el navegador enviará los datos a `PrintEnv`, obteniendo un resultado como el de la ilustración.

precio	334,44
dni	77232917-K
nombre	Nito del Bosque, Elena
email	elena@dot.com

- Por curiosidad, puede usarse el NIE como dato, sustituyendo la primera letra: si es un X, cambiar por 0; si es una Y, cambiar por 1; si es una Z, cambiar por un 2.

5.2 Funciones adicionales: fecha automática

En una variable oculta denominada **fecha**, que deberemos añadir al formulario, debe colocarse una información que informe sobre el momento de envío del formulario, con el valor que ofrece la función Date, como *"Tue Feb 13 2018 18:04:33 GMT+0100"*.

tarjeta	Amex
numTarj1	214748364700
numTarj2	214748364700
fecha	Tue Feb 13 2018 18:04:33 GMT+0100

Ambas funciones adicionales (letra del NIF y fecha) únicamente deberán llevarse a cabo si se superan las comprobaciones que se mencionan en los dos apartados siguientes, e inmediatamente antes del envío.

5.3 Comprobaciones: letra del NIF

La letra del NIF pierde todo su interés si se calcula automáticamente. Su propósito es acompañar a un DNI para verificar que no hay ningún error en el mismo.

Para respetar esa utilidad hemos de replantear este elemento: el campo DNI (variable `dni`) pasa a llamarse NIF (variable `nif`), y deberemos comprobar que el NIF introducido es válido, por lo que llamamos a la función **comprobarNIF()**. Podemos considerar todos los casos de error (cada uno se apoya en el anterior):

1. ¿Se ha introducido algún valor? Fragmento significativo de código:

```
if (f.nif.value == "") {
    alert("El NIF no puede estar vacío");
    return false;
}
```

2. ¿Ese valor se corresponde con un número de 8 cifras? (seguido por un guión y una letra). Lo podemos simplificar como número de 8 cifras y 2 símbolos al final. Fragmento significativo de código:

```
if ((f.nif.value.length != 10) || !(esEntero(f.nif.value.substr(0, 8))) {
    alert("El NIF no se corresponde con el formato esperado DNI-letra");
    return false;
}
```

Para la implementación de `esEntero()` se puede optar por varias alternativas. Destaco:

- a) Usar el operador `isNaN()`, que devuelve TRUE en caso de que el argumento no tenga el formato de un número.
- b) Usar `parseInt`.
- c) Recorrer la cadena para comprobar cada elemento.
- d) Aplicar una expresión regular JavaScript.

3. ¿La letra coincide con la que calcularíamos para ese número?. Fragmento significativo de código, suponiendo la misma variable letras que en calculaLetraNIF:

```
u = letras[(f.nif.value.substr(0, 8)) % 23];
if (!f.nif.value.endsWith(u)) {
    alert("La letra del NIF no coincide (debería ser "+u+"");
    return false;
}
```

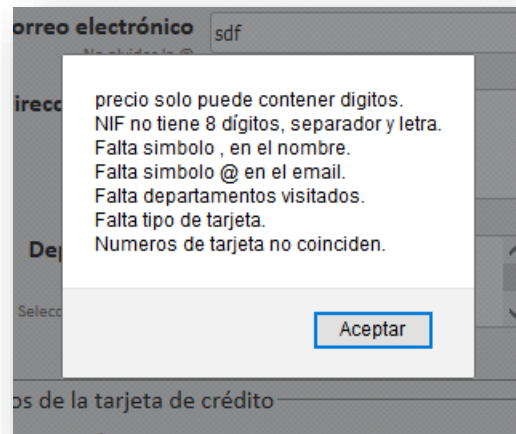
Si se superan todas estas comprobaciones, podemos optar *provisionalmente*⁸ por informar mediante un `alert` y devolver `true`.

```
alert("Comprobación del NIF superada");
return true;
```

5.4 Comprobaciones: requisitos de la solución

El formulario que se ha desarrollado debe mejorarse para garantizar que los datos introducidos cumplen las siguientes **condiciones**:

1. Todos los campos tienen valor.
2. El precio por unidad es un valor entero sin signo.
3. El NIF será un entero sin signo de 8 cifras, seguido de un separador (p.ej. un guión) y la letra correspondiente.
4. Hay una única coma separando el nombre y los apellidos, y debe tener texto a ambos lados de la coma.
5. Cambiando coma por arroba, se exige la misma característica para la dirección de correo electrónico.
6. Se ha seleccionado al menos un departamento, pero podrían ser varios.
7. Se debe seleccionar un tipo de tarjeta de crédito, exactamente.
8. El número de la tarjeta es un entero de 12 cifras, que NO puede comenzar por 0.
9. Los dos números de tarjeta introducidos deben coincidir.



Además, debe añadirse una **variable oculta** "fecha" que informe sobre el momento en que se envió el formulario.

⁸ en la versión final sólo se informa sobre errores

Si no se cumple alguna de las condiciones anteriores tras cumplimentar los datos, **al pulsar el botón de envío**, deberemos avisar con una ventana de alerta, informando sobre todos los errores cometidos.

- Esto debería permitir su modificación, puesto que los contenidos del formulario todavía se encontrarán presentes.
- Este paso se realizará **tantas veces como sea necesario**, al pulsar el botón de envío, mientras quede algún error por resolver.

Como punto de partida tenemos disponibles (en PoliformaT) estos dos ficheros:

- `formToCheckDEW.html`: formulario HTML que hay que validar. Completo y “apuntando” al programa Printenv en `memex.dsic.upv.es`.
- `tsr.css`: fichero con los estilos utilizados en el formulario. Completo.

Como resultado entregable de esta parte relacionada con el formulario, deberás crear dos páginas:

1. La primera es una *aproximación* a la solución sin emplear JavaScript ni el atributo `pattern`, usando las capacidades restantes de HTML5 para verificar valores de formularios. Unas propiedades se podrán garantizar pero otras no. Por ejemplo:
 - El tipo `number` no permite especificar su longitud (en cifras) y es incompatible con el tipo `password`
 - No hay un tipo adecuado para el NIF que pueda colocarse como restricción
 - No se pueden comparar entre sí diferentes elementos de un formulario

La página resultante se denominará `casiformToCheckDEW.html`, y su funcionamiento y limitaciones deben ser descritos en la memoria de la práctica.

2. La **segunda**, de la que se han mostrado ya todos los detalles, NO se basa en la anterior y Sí que emplea JavaScript. Se denominará `mi formToCheckDEW.html` y su funcionamiento debe ser descrito en la memoria de la práctica.

6 TU SERVIDOR VIRTUAL Y TOMCAT

El servidor de aplicaciones Tomcat se encuentra preinstalado en los servidores virtuales del portal, en el directorio `/root/tomcat`, incluyendo documentación adicional a la habitual de Tomcat. Los directorios y ficheros más importantes a tener en cuenta para estas sesiones son:

- **bin** directorio donde se encuentran los dos scripts que vamos a utilizar para iniciar y detener el servidor: `startup.sh` y `shutdown.sh`
- **webapps** directorio donde se ubican las aplicaciones web desplegadas en Tomcat.
 - Dentro de esta carpeta colocaremos un directorio `dew1` para esta práctica.

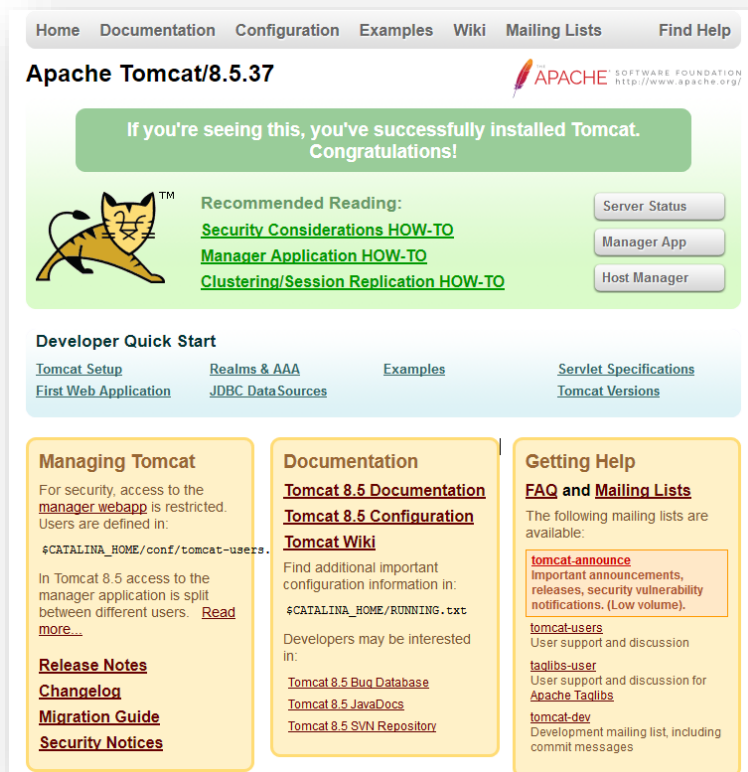
En resumen, `/root/tomcat/webapps/dew1` será la ruta del directorio donde dejaremos todo el material de esta práctica

- **logs** directorio donde se encuentran los ficheros que registran los eventos más importantes (se puede determinar en la configuración) durante la ejecución de Tomcat.
- **conf** directorio donde se guarda la configuración de Tomcat.
- **lib** directorio con las bibliotecas utilizadas por el servidor.

Para comprobar que el proceso ha concluido correctamente hay que seguir los siguientes pasos:

1. Arrancar Tomcat (ejecuta el script `/root/tomcat/bin/startup.sh`).
2. Abrir un navegador web.
3. Acceder al URL `http://dew-tulogin-1819.dsic.cloud:8080`. Por defecto Tomcat está configurado para atender al puerto 8080.

Si todo ha funcionado correctamente debes ver en la pantalla del navegador la página principal por defecto de Tomcat (ilustración de la derecha).



6.1 Habilitar red, arrancar, detener

En la configuración inicial de *otras* máquinas virtuales el **cortafuegos** impide el acceso a la mayoría de puertos del equipo desde el exterior. En nuestro caso ya se ha previsto esta eventualidad y se ha resuelto con antelación.

- Para habilitar el acceso al servidor Tomcat (puerto 8080) *en otros sistemas* deberás ejecutar instrucciones similares a las siguientes:

```
firewall-cmd --permanent --add-port=8080/tcp
firewall-cmd --reload
```

- Posiblemente también necesites, *en otros sistemas*, habilitar el acceso al servidor Tomcat en modo seguro, cambiando 8080 por 8443 en el ejemplo anterior.

Observa que el contenedor **no** se inicia automáticamente con el arranque del equipo.

Arrancar Tomcat con:

```
[root@localhost ]# /root/tomcat/bin/startup.sh
Using CATALINA_BASE:   /root/tomcat
Using CATALINA_HOME:   /root/tomcat
Using CATALINA_TMPDIR: /root/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:        /root/tomcat/bin/bootstrap.jar:/root/tomcat/bin/tomcat-juli.jar
Tomcat started
```

A partir de este momento el servidor está accesible por HTTP en el puerto 8080. Debes probarlo con un navegador en cualquier equipo con acceso a la VPN, habitualmente desde tu escritorio virtual.

Detener Tomcat con:

```
[root@localhost ]# /root/tomcat/bin/shutdown.sh
Using CATALINA_BASE:   /root/tomcat
Using CATALINA_HOME:   /root/tomcat
Using CATALINA_TMPDIR: /root/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:        /root/tomcat/bin/bootstrap.jar:/root/tomcat/bin/tomcat-juli.jar
```

7 ENTREGA

En PoliformaT encontraréis una tarea (**Bloque 1 laboratorio**) para depositar el material desarrollado. La fecha de cierre de la tarea es el 10 de Marzo. Las condiciones reales de la entrega deben garantizar que el material aportado funciona simplemente copiándolo en el directorio /root/tomcat/webapps/dew1 de vuestro servidor virtual de portal.

Desgraciadamente, dada la naturaleza de estos materiales, es **muy sencillo obtener una copia** de otros porque el acceso a esos servidores de portal no se ha limitado adecuadamente (aprenderemos a hacerlo en la segunda práctica). Por ello:

- Debéis cambiar la contraseña de root por otra nueva, bajo vuestra responsabilidad.
- No debéis dejar vuestro servidor virtual en funcionamiento cuando no lo uséis.

7.1 Materiales a aportar

El material que se debe adjuntar en la entrega será un fichero comprimido (zip, gzip) que debe incluir:

1. Los materiales HTML/CSS/JavaScript que son resultado de las prácticas: dew1.html, tsr.css, p2.html (junto con las ilustraciones y recursos de los que depende), ej1.html (en versiones Bootstrap con estética *Superhero* y *Sketchy*, con sus directorios respectivos para las hojas de estilo), casiformToCheckDEW.html, miformToCheckDEW.html, ej1Fondamento.html y ej1Montserrat.html
 - Esos materiales deben consistir en código fuente directamente “consumible” por el navegador. No sirven capturas, PDFs ni similares.
 - Debe estar organizado de manera que encaje con la página dew1.html cuyo código se muestra en el anexo 4.
 - Es el mismo archivo dew1.tar.gz mencionado en el apartado 1.1.
 - Su "instalación" debe consistir únicamente en descomprimir el archivo en el directorio webapps/dew1 del servidor virtual.
 - No debe requerir ningún otro ajuste adicional.
 - Comprobadlo porque es extremadamente importante.
2. La memoria, en formato PDF (**memoriaDEW-1.pdf**), formada por:
 - Una **portada** que incluya la relación de los 2 alumnos que han intervenido (apellidos, nombre y email en la UPV) y el grupo de laboratorio al que pertenecen
 - Identificación de la máquina virtual, tipo dew-tulogin-1819.dsic.cloud, en la que se ha dejado los materiales, accesibles en el servidor de web, por el puerto 8080.
 - Una **tabla resumen** que indique, para cada actividad, el nivel de consecución de los objetivos, distinguiendo entre: completo (3), bastante (2), algo (1) y nada (0), como una *autocalificación*.
 - Una **descripción de cada actividad**, con un nivel de detalle suficiente para mostrar qué trabajo se ha realizado, prescindiendo del enunciado
 - Indicar, si las hay, las diferencias entre la solución propuesta y el resultado deseable (Sketchy, Superhero, miformToCheckDEW.html)
 - Descripción (más detallada) de las limitaciones de casiformToCheckDEW.html y del funcionamiento de miformToCheckDEW.html

8 ANEXO 0. INTRODUCCIÓN Y PREPARACIÓN DE TOMCAT

Como ya sabes, los laboratorios del DSIC permiten la realización de prácticas con diferentes configuraciones en cuanto a aplicaciones y sistemas operativos. Por ello pueden convivir recursos de varias asignaturas que no siempre son intercambiables.

En relación con DEW nos interesa:

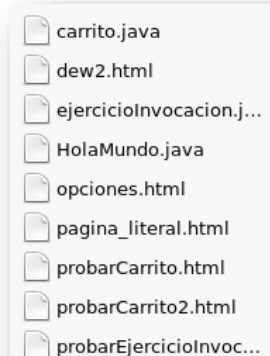
1. En nuestro equipo de escritorio: Linux-CentOS 7 de 64 bits, navegador Firefox, entorno de desarrollo Eclipse Oxygen para JavaEE, Java 1.8

No usaremos contenedores Tomcat en los equipos de escritorio porque la reserva del puerto necesario es incompatible con los escritorios virtuales, en los que los equipos físicos se comparten entre múltiples usuarios (sólo el primero puede *apropiarse* del puerto!)

2. En nuestro servidor virtual: Linux-CentOS 7 de 64 bits, contenedor Apache Tomcat 8.5.X, Java 1.8
 - El contenedor de servlets Tomcat 8.5 es compatible con las especificaciones 2.x y 3.1. Nuestro objetivo es limitarnos a la especificación de **servlets 2.5**.
 - Puedes consultar el API de dicha versión 2.5 en <http://memex.dsic.upv.es:8080/api>, que contiene una copia de la especificación original.
 - Es **conveniente detener nuestro servidor virtual** al finalizar una sesión para no congestionar recursos de la infraestructura.

A lo largo de la práctica se utilizarán distintos ficheros contenidos en asigDSIC/ETSINF/dew/labo1, algunos de los cuales se enumeran a la derecha.

Por último, os informo de que no hay forma sencilla de conocer el **espacio libre disponible** para cada alumno dentro del sistema de ficheros compartido. Esto puede ser especialmente preocupante porque Eclipse puede consumir mucho espacio, y el problema se agrava si mantenemos archivos procedentes del cuatrimestre anterior.



Por ello se sugiere seguir estas pautas:

- 1) Al iniciar la sesión ejecutar en una ventana de órdenes "du -xsh \$HOME", que indica cuánto se está ocupando en el espacio compartido
- 2) Determinar si hace falta aligerar algo antes de comenzar la sesión. Compara el espacio ocupado con el límite de 400MB⁹
- 3) Si es necesario recuperar espacio, los candidatos más habituales son la Papelera (no siempre se vacía) y la cache del navegador (puede borrarse desde el menú Editar del navegador).

⁹ Suele variar en cada curso

Es importante conocer que la falta de espacio remoto no está bien controlada por las aplicaciones, que pueden fallar de forma *poco amable*.

8.1 Configurando HTTPS en Tomcat (informativo)

Todo este apartado describe la configuración YA REALIZADA en los servidores de DEW.

La evolución en la web va descubriendo problemas a los que se intenta encontrar solución. Un porcentaje muy elevado de esos problemas están relacionados con la inseguridad de Internet, que motiva el uso de **la variante segura de HTTP**. Si el certificado del servidor seguro es expedido por una entidad reconocida, podremos evitar problemas de suplantación; y por otro lado, como el protocolo seguro incluye cifrado, mejorará la confidencialidad de la comunicación.

En [¹⁰ https://tomcat.apache.org/tomcat-8.5-doc/ssl-howto.html](https://tomcat.apache.org/tomcat-8.5-doc/ssl-howto.html) puedes obtener información sobre los pasos necesarios para configurar Tomcat de manera que pueda mantener **conexiones seguras**. Destacamos:

1. Preparamos la parte criptográfica para SSL: generamos el par de claves necesarias mediante la aplicación **keytool**, incluida en las distribuciones de la MV Java:

```
keytool -genkey -alias tomcat -keyalg RSA
```

- Habrá que colocar una contraseña (*laclave*)

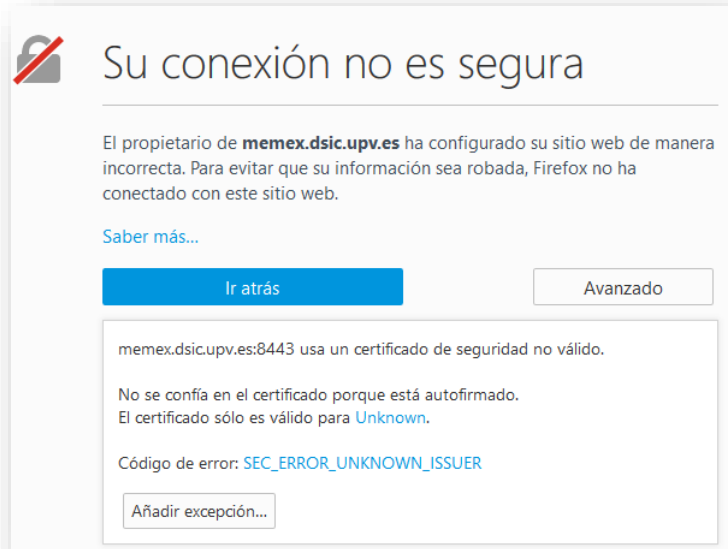
```
[root@dew-tulogin-1819]# cd tomcat/conf
[root@dew-tulogin-1819]# keytool -genkey -alias tomcat -keyalg RSA
Introduzca la contraseña del almacén de claves: <introducir la clave>
Volver a escribir la contraseña nueva: <introducir la clave>
¿Cuáles son su nombre y su apellido? [Unknown]: <INTRO>
¿Cuál es el nombre de su unidad de organización? [Unknown]: <INTRO>
¿Cuál es el nombre de su organización? [Unknown]: <INTRO>
¿Cuál es el nombre de su ciudad o localidad? [Unknown]: <INTRO>
¿Cuál es el nombre de su estado o provincia? [Unknown]: <INTRO>
¿Cuál es el código de país de dos letras de la unidad? [Unknown]: <INTRO>
¿Es correcto CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown? [no]: si
Introduzca la contraseña de clave para <tomcat>
(<INTRO> si es la misma contraseña que la del almacén de claves)
```

2. En el archivo `server.xml` de la configuración de Tomcat descomentamos y modificamos la entrada del conector "SSL HTTP/1.1" para que quede así:

```
<Connector protocol="org.apache.coyote.http11.Http11NioProtocol"
  port="8443" maxThreads="150" SSLEnabled="true"
  scheme="https" secure="true" keystorePass="laclave"
  clientAuth="false" sslProtocol="TLS" />
```

¹⁰ también incluido como documentación en nuestros servidores

3. Una vez preparado, el servidor será accesible también en la URL `https://localhost:8443/`
 - Se quejará por no disponer de un certificado **oficial**.
4. Para poder acceder desde fuera al servidor, habilitamos el puerto 8443¹¹ en el cortafuegos, análogamente al caso del puerto 8080 en la práctica 1.



Tras reiniciar el contenedor, el siguiente URL debería¹² funcionar:

`https://dew-tulogin-1819.dsic.cloud:8443/dew1/dew1.html`

¹¹ El puerto 443 oficial requiere privilegios

¹² Si obligamos al navegador a confiar en el certificado *artesano* del servidor

9 ANEXO 1. EJ1.HTML: FUENTE Y VISUALIZACIÓN (SIN ESTILOS)

Accesible en <http://memex.dsic.upv.es:8080/dew1/ej1.html>

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="revised" content="13/02/2018" />
  <title>Práctica 1: HTML</title>
</head>
<body>

<h1>Todo página debería tener un título.</h1>

<p>Claro que después habrá que poner el texto normal.</p>
<hr />
<p class="nota">Recuerda que siempre puedes ver el código fuente de una
página
HTML desde el navegador.</p>
<p class="nota">Tampoco debes olvidar que se pueden guardar las
páginas
en ficheros, para luego editarlas si es necesario.</p>
<p class="nota">Cuando uses Eclipse, el editor de páginas web te permite
simultáneamente la vista del código fuente con su visualización.</p>
<p class="nota">Cuando uses un editor de texto plano, te aconsejo que mantengas
abiertos tanto el navegador como el editor
elegido. De esta manera podrás observar inmediatamente las modificaciones
que realices en la página.</p>
<p class="nota">Cuidado con los derechos de autor si deseas emplear imágenes
de terceros. En Wikimedia Commons (<a
href="http://commons.wikimedia.org/wiki/Portada"></a>) dispones de un
código logo de imágenes y otros recursos multimedia libres.</p>

<h2>Los títulos de nivel inferior destacan aspectos relativamente menos
importantes.</h2>

<h3>Los de nivel 3 destacan todavía menos.</h3>

<h4>Y no debes pasar de los de nivel 4.</h4>
<p>Observarás que no destacan demasiado respecto al texto plano.</p>
<h2>Es bueno practicar con las listas.</h2>
<p>La primera será; una lista no ordenada para enumerar:</p>

<ul>
<li>Lo que debería aprender</li>
<li>Lo que ya sé hacer</li>
<li>Lo que se supone que debería conocer</li>
</ul>

<hr />
<p>Claro que los elementos que tienen un orden deben ser indicados de forma
diferente:</p>
<ol>
<li>Batir los huevos</li>
<li>Añadir el tomate</li>
<li>Salpimentar al gusto</li>
```

```

<li>Freir a fuego fuerte</li>
<li>Retirar cuando comience a dorarse</li>
<li>Servir a temperatura ambiente</li>
</ol>

<hr />
<p>Hay otros tipos de lista que son m&acutes; complicadas, pero su utilidad
estriba en:</p>
<dl>
<dt>Conveniencia</dt>
<dd>Puesto que hay momentos en los que nos interesa separar el t&eacutes;mino
a definir de se definici&ocutes;n</dd>
<dt>Claridad</dt>
<dd>Dado que esta separaci&ocutes;n destaca y separa el t&eacutes;mino a
definir</dd>
<dt>Versatilidad</dt>
<dd>Podr&iacutes;amos construir un programa que calculase un &iacutes;ndice de
t&eacutes;minos seleccionando los que se han definido</dd>
</dl>

<hr />
<p>Desde luego que tenemos la posibilidad de elegir los bullets (tambi&eacutes;n
denominados vi&ntilde;etas en algunos programas) o el sistema de numeraci&ocutes;n
a aplicar (p.ej. alfab&eacutes;tico), pero esas capacidades a&ntilde;adidas
son un extra en este momento. Recuerda que se obtienen mediante atributos
aplicados a las listas.</p>
<hr />
<h2>No s&ocutes;lo de listas vive el hombre ... tambi&eacutes;n se necesita
tener tablas</h2>
<p>Es dif&iacutes;cil encontrar argumentos nuevos para defender la existencia
de esta estructura. Un ejemplo sencillo podr&iacutes;a ser un calendario
de la semana laboral:</p>

<table>
<tr><td>Lunes</td><td>Martes</td><td>Mi&eacutes;rcoles</td><td>Jueves</td><td>Viernes</td></tr>
<tr><td>26</td><td>27</td><td>28</td><td>01</td><td>02</td></tr>
</table>
<p>Pero quiz&acutes; esto fuera demasiado obvio ..., &quest;qu&eacutes; tal
una tabla no homog&eacutes;nea sin bordes?</p>

<hr />
<h2>Algo de formularios</h2>
<p>El primer formulario que vamos a crear s&ocutes;lo nos sirve para comprobar
que funciona. Hay un programita <var>Printenv</var> que nos indicar&acutes;
qu&eacutes; ha recibido el servidor al pulsar el bot&ocutes;n de env&iacutes;o.</p>

<form action="http://memex.dsic.upv.es:8080/dew1/Printenv" method="get">
<p><input type="submit" value="Pulsa aqu&iacutes;, valiente" /></p></form>
<hr />
<p>Empezamos a complicar el asunto colocando algunos elementos extra.</p>
<p>Todo lo que sigue, hasta el bot&ocutes;n <var>Segunda prueba</var>, forma parte
del mismo formulario:</p>
<form action="http://memex.dsic.upv.es:8080/dew1/Printenv" method="get">
<p>Un texto: <input type="text" name="var1" size=30 /></p>
<p>&Acutes;rea de texto: <textarea name="var2" rows="5" cols="25"></textarea></p>
<p>Opciones compatibles (independientes entre s&iacutes;):<br />
<input type="checkbox" name="var3" value="yes" />No me pinches,<br />
<input type="checkbox" name="var4" value="yes" />te lo advierto,<br />

```

```

<input type="checkbox" name="var5" value="yes" />que me enfado<br />
<input type="checkbox" name="var6" value="yes" />y te reviento.</p>
<p>Se puede elegir una de las siguientes opciones (la segunda
tiene ya una marca por omisión):<br />
<input type="radio" name="var7" value="1" />Primera opción.<br />
<input type="radio" name="var7" value="2" checked="checked" />Segunda
opción.<br />
<input type="radio" name="var7" value="3" />Tercera opción.</p>
<p><input type="submit" value="Segunda prueba" /></p>
</form>
<hr />
<p>Comencemos el tercer formulario. Aquí usaremos listas y menús
desplegables</p>
<form action="http://memex.dsic.upv.es:8080/dew1/Printenv" method="get">
<p>Dime de dónde vienes:<br />
<select name="var8">
<option value="valcap">de Valencia capital</option>
<option value="valprov">de la provincia de Valencia</option>
<option value="casprov">de la provincia de Castellón</option>
<option value="aliprov">de la provincia de Alicante</option>
<option value="otros">no, vengo de otro lugar</option>
</select></p>
<p>En ocasiones uno puede elegir varios valores si no son incompatibles. Hace falta
pulsar la tecla de mayúsculas junto con el botón del ratón si
se desea marcar varias opciones</p>
<p>Dime por qué estudias Informática: <select name="var9"
multiple="multiple" size="3">
<option value="noestudioinf">yo no estudio Informática</option>
<option value="megusta">porque me gusta</option>
<option value="masoquista">porque soy masoquista</option>
<option value="nosabiaelegir">porque no sabía qué elegir</option>
<option value="millonario">porque quiero ser millonario</option>
<option value="famoso">porque quiero ser famoso</option>
<option value="artista">¡¡¡, quiero ser artista</option>
<option value="sacadme">¡¡¡sacadme de aquí!</option>
<option value="norollos">no estoy para rollos</option>
<option value="noestudio">¿estudiar?, ¿qué es eso?</option>
</select></p>

<p><input type="submit" value="Tercer formulario" /></p>
</form>
<hr />
<p>¡... y esto ha sido todo por hoy!</p>

</body>
</html>

```

Toda página debería tener un título.


Claro que después habrá que poner el texto normal.

Recuerda que siempre puedes ver el código fuente de una página HTML desde el navegador.

Tampoco deberías olvidar que se pueden guardar las páginas en ficheros, para luego editarlas si es necesario.

Cuando uses Eclipse, el editor de páginas web te permite simultáneas la vista del código fuente con su previsualización

Cuando uses un editor de texto llano, te aconsejo que mantengas abiertos tanto el navegador como el editor elegido. De esta manera podrás observar inmediatamente las modificaciones que realices en la página.

Cuidado con los derechos de autor si deseas emplear imágenes de terceros. En Wikimedia Commons () dispones de un catálogo de imágenes y otros recursos multimedia libres.

Los títulos de nivel inferior destacan aspectos relativamente menos importantes.

Los de nivel 3 destacan todavía menos.

Y no deberías pasar de los de nivel 4.

Observarás que no destacan demasiado respecto al texto llano.

Es bueno practicar con las listas.

La primera será una lista no ordenada para enumerar:

- Lo que debería aprender
- Lo que ya sé hacer
- Lo que se supone que debería conocer

Claro que los elementos que tienen un orden deben ser indicados de forma diferente:

1. Batir los huevos
2. Añadir el tomate
3. Salpimentar al gusto
4. Freír a fuego fuerte
5. Retirar cuando comience a dorarse
6. Servir a temperatura ambiente

Hay otros tipos de lista que son más complicadas, pero su utilidad estriba en:

Conveniencia

Puesto que hay momentos en los que nos interesa separar el término a definir de su definición

Claridad

Dado que esta separación destaca y separa el término a definir

Versatilidad

Podríamos construir un programa que calculase un índice de términos seleccionando los que se han definido

Desde luego que tenemos la posibilidad de elegir los bullets (también denominados viñetas en algunos programas) o el sistema de numeración a aplicar (p.ej. alfabético), pero esas capacidades añadidas son un extra en este momento. Recuerda que se obtienen mediante atributos aplicados a las listas.

No sólo de listas vive el hombre ... también se necesita tener tablas

Es difícil encontrar argumentos nuevos para defender la existencia de esta estructura. Un ejemplo sencillo podría ser un calendario de la semana laboral:

Lunes	Martes	Miércoles	Jueves	Viernes
26	27	28	01	02

Pero quizá esto fuera demasiado obvio ..., ¿qué tal una tabla no homogénea sin bordes?

Algo de formularios

El primer formulario que vamos a crear sólo nos sirve para comprobar que funciona. Hay un programita *Printenv* que nos indicará qué ha recibido el servidor al pulsar el botón de envío.

Pulsa aquí, valiente

Empezamos a complicar el asunto colocando algunos elementos extra.

Todo lo que sigue, hasta el botón *Segunda prueba*, forma parte del mismo formulario:

Un texto:

Área de texto:

Opciones compatibles (independientes entre sí):

- ☐ No me pinches,
- ☐ te lo advierto,
- ☐ que me enfado
- ☐ y te reviento.

Sólo podrás elegir una de las siguientes opciones (la segunda tiene ya una marca por omisión):

- ☐ Primera opción.
- ☒ Segunda opción.
- ☐ Tercera opción.

Segunda prueba

Comencemos el tercer formulario. Aquí usaremos listas y menús desplegables

Dime de dónde vienes:

En ocasiones uno puede elegir varios valores si no son incompatibles. Hace falta pulsar la tecla de mayúsculas junto con el botón del ratón si se desea marcar varias opciones

Dime por qué estudias Informática:

Tercer formulario

¡... y esto ha sido todo por hoy!

10 ANEXO 2. ESTILO (AZUL) Y ASPECTO RESULTANTE

Archivo **estiloAzul.css** (crea antes un directorio css), accesible en el URL <http://memex.dsic.upv.es:8080/dew1/css/estiloAzul.css>

```
p      { font-family: sans-serif;}
p.nota { color:white; background:black;}
ul.li  { color:white; background:red;
        font-family:sans-serif;}
ol.li  { color:white; background:blue; font-family:Comic, sans-serif;}
dt      { color:white; background:blue; font-size:16px;}
td      { color:darkred; background:white; font-family:helvetica, serif;}
table   { align:center;}
body    { background-color:#cfddef;
        font-family: serif;
        color:darkblue;
        margin-left:8px;}

h3      { color:darkred;
        margin-top:8px;
        font-size:24px;
        font-weight:bold;
        font-family: serif;
        text-shadow:4px 4px 4px #aaa;
        font-variant:small-caps;
        font-family: fantasy;}

h2      { color:darkgreen;
        margin-top:8px;
        font-size:30px;
        font-weight:bold;
        text-shadow:4px 4px 4px #aaa;
        font-variant:small-caps;
        font-family: fantasy;}

h1      { color:darkblue;
        margin-top:10px;
        font-size:36px;
        font-weight:bold;
        font-family: serif;
        text-shadow:4px 4px 4px #aaa;
        font-variant:small-caps;
        font-family: fantasy;}

I       { font-weight:bold;
        color:azure;}

a:link  { font-size:18px;
        text-decoration:none;
        color:green;
        font-weight:bold;
        font-family: fantasy, sans-serif;}
a:visited { font-size:12px;
        text-decoration:none;
        color:#AAAAAA;
        font-family: fantasy, sans-serif;}
```

Modificamos **ej1.html** (resultado es **ej1Estilo.html**) para referenciar el estilo (accesible en <http://memex.dsic.upv.es:8080/dew1/ej1Estilo.html>)

```
<head>
...
<link rel="stylesheet" type="text/css" href="css/estiloAzul.css" />
</head>
```


TODA PÁGINA DEBERÍA TENER UN TÍTULO.


Claro que después habrá que poner el texto normal.

Recuerda que siempre puedes ver el código fuente de una página HTML desde el navegador.

Tampoco deberías olvidar que se pueden guardar las páginas en ficheros, para luego editarlas si es necesario.

Cuando uses Eclipse, el editor de páginas web te permite simultáneas la vista del código fuente con su previsualización

Cuando uses un editor de texto llano, te aconsejo que mantengas abiertos tanto el navegador como el editor elegido. De esta manera podrás observar inmediatamente las modificaciones que realices en la página.

Cuidado con los derechos de autor si deseas emplear imágenes de terceros. En Wikimedia Commons () dispones de un catálogo de imágenes y otros recursos multimedia libres.

LOS TÍTULOS DE NIVEL INFERIOR DESTACAN ASPECTOS RELATIVAMENTE MENOS IMPORTANTES.

LOS DE NIVEL 3 DESTACAN TODAVÍA MENOS.

Y no deberías pasar de los de nivel 4.

Observarás que no destacan demasiado respecto al texto

ES BUENO PRACTICAR CON

La primera será una lista no ordenada para enumerar:

- Lo que debería aprender
- Lo que ya sé hacer
- Lo que se supone que debería conocer

Claro que los elementos que tienen un orden deben ser

1. Batir los huevos

2. Añadir el tomate
3. Salpimentar al gusto
4. Freír a fuego fuerte
5. Retirar cuando comience a dorarse
6. Servir a temperatura ambiente

Hay otros tipos de lista que son más complicadas, pero su utilidad estriba en:

Conveniencia

Puesto que hay momentos en los que nos interesa separar el término a definir de su definición

Claridad

Dado que esta separación destaca y separa el término a definir

Versatilidad

Podríamos construir un programa que calculase un índice de términos seleccionando los que se han definido

Desde luego que tenemos la posibilidad de elegir los bullets (también denominados viñetas en algunos programas) o el sistema de numeración a aplicar (p.ej. alfabético), pero esas capacidades añadidas son un extra en este momento. Recuerda que se obtienen mediante atributos aplicados a las listas.

NO SÓLO DE LISTAS VIVE EL HOMBRE ... TAMBIÉN SE NECESITA TENER TABLAS

Es difícil encontrar argumentos nuevos para defender la existencia de esta estructura. Un ejemplo sencillo podría ser un calendario de la semana laboral:

Lunes	Martes	Miércoles	Jueves	Viernes
26	27	28	01	02

Pero quizá esto fuera demasiado obvio ..., ¿qué tal una tabla no homogénea sin bordes?

ALGO DE FORMULARIOS

El primer formulario que vamos a crear sólo nos sirve para comprobar que funciona. Hay un programita *Printenv* que nos indicará qué ha recibido el servidor al pulsar el botón de envío.

Empezamos a complicar el asunto colocando algunos elementos

Todo lo que sigue, hasta el botón *Segunda prueba*, forma

Un texto:

Área de texto:

Opciones compatibles (independientes entre sí):

- ☐ No me pinches,
☐ te lo advierto,
☐ que me enfado
☐ y te reviento.

Sólo podrás elegir una de las siguientes opciones (la selección es

- ☐ Primera opción.
☒ Segunda opción.
☐ Tercera opción.

Comencemos el tercer formulario. Aquí usaremos listas y menús desplegables

Dime de dónde vienes:

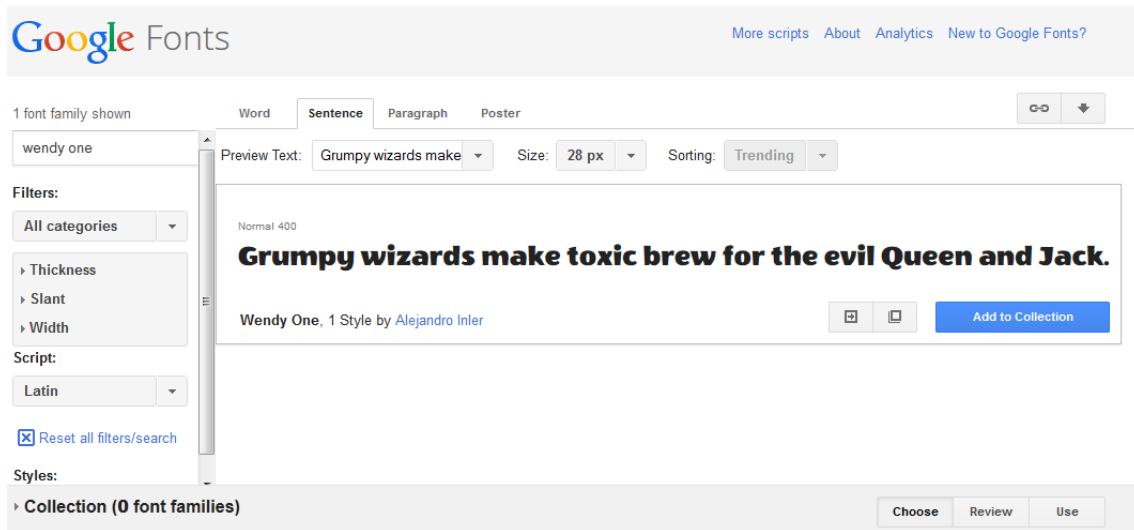
En ocasiones uno puede elegir varios valores si no son incompatibles. Hace falta pulsar la tecla de mayúsculas junto con el botón del ratón si se desea marcar varias opciones

Dime por qué estudias Informática:

¡... y esto ha sido todo por hoy!

11 ANEXO 3. TIPOGRAFÍA EN GOOGLE WEB FONTS

Información en <http://www.google.com/fonts>. Usaremos los tipos *Fondamento* y *Montserrat*.



Modificaremos la hoja de estilo anterior (**estiloAzul.css**) para obtener una versión que utilice la primera tipografía (*Fondamento*, llamaremos **estiloFondamento.css** y **ej1Fondamento.html** a la nueva hoja y el nuevo documento), y otro para la segunda (*Montserrat*, con **estiloMontserrat.css** y **ej1Montserrat.html** respectivamente).

11.1 Aspectos destacables (ejemplos con Fondamento):

En la hoja de estilos se “consume” la tipografía como `font-family: 'Fondamento', cursive`. Debe aplicarse a todos los tipos de letra.

En la página HTML la cabecera debe incluir una sentencia de importación, tal y como indica la documentación de Google Web Fonts

```
<link href='https://fonts.googleapis.com/css?family=Fondamento' rel='stylesheet' type='text/css' />
```

... antes de la importación de nuestra propia hoja de estilos `estiloFondamento.css`

Recuerda que puedes ver los resultados deseados en <http://memex.dsic.upv.es:8080/dew1/ej1Fondamento.html> y en <http://memex.dsic.upv.es:8080/dew1/ej1Montserrat.html>

TODA PÁGINA DEBERÍA TENER UN TÍTULO.

Claro que después habrá que poner el texto normal.

Recuerda que siempre puedes ver el código fuente de una página HTML desde el navegador.

Tampoco deberías olvidar que se pueden guardar las páginas en ficheros, para luego editarlas si es necesario.

Cuando uses Eclipse, el editor de páginas web te permite simultáneas la vista del código fuente con su previsualización

Cuando uses un editor de texto llano, te aconsejo que mantengas abiertos tanto el navegador como el editor elegido. De esta manera podrás observar inmediatamente las modificaciones que realices en la página.

Cuidado con los derechos de autor si deseas emplear imágenes. Dispones de un catálogo de imágenes y otros recursos multimedios.

LOS TÍTULOS DE NIVEL INFERIOR SON RELATIVAMENTE MENOS IMPORTANTES

LOS DE NIVEL 3 DESTACAN TODAVÍA MÁS

Y no deberías pasar de los de nivel 4.

Observarás que no destacan demasiado respecto al texto llano.

ES BUENO PRACTICAR CON LAS LISTAS

La primera será una lista no ordenada para enumerar:

- Lo que debería aprender
- Lo que ya sé hacer
- Lo que se supone que debería conocer

Claro que los elementos que tienen un orden deben ser indicados.

1. Batir los huevos
2. Añadir el tomate
3. Salpimentar al gusto
4. Freír a fuego fuerte
5. Retirar cuando comience a dorarse
6. Servir a temperatura ambiente

Es difícil encontrar argumentos nuevos para defender la existencia de esta estructura. Un ejemplo sencillo podría ser un calendario de la semana laboral:

Lunes	Martes	Miércoles	Jueves	Viernes
26	27	28	01	02

Pero quizá esto fuera demasiado obvio ..., ¿qué tal una tabla no homogénea sin bordes?

ALGO DE FORMULARIOS

El primer formulario que vamos a crear sólo nos sirve para comprobar que funciona. Hay un programita *Printenv* que nos indicará qué ha recibido el servidor al pulsar el botón de envío.

Pulsa aquí, valiente

Empezamos a complicar el asunto colocando algunos elementos extra.

Todo lo que sigue, hasta el botón *Segunda prueba*, forma parte del mismo formulario:

Un texto:

Área de texto:

Opciones compatibles (independientes entre sí):

- ☐ No me pinches,
☐ te lo advierto,
☐ que me enfado
☐ y te reviento.

Sólo podrás elegir una de las siguientes opciones (la segunda tiene ya una marca por omisión):

- ☐ Primera opción.
☒ Segunda opción.
☐ Tercera opción.

Segunda prueba

Comencemos el tercer formulario. Aquí usaremos listas y menús desplegables

Dime de dónde vienes:

En ocasiones uno puede elegir varios valores si no son incompatibles. Hace falta pulsar la tecla de mayúsculas junto con el botón del ratón si se desea marcar varias opciones

Dime por qué estudias Informática:

Tercer formulario

... y esto ha sido todo por hoy!

12 ANEXO 4. DEW1.HTML

Es el documento director de todo el entregable, permitiendo unificar la organización del material producido en la práctica. Dispones de una copia del mismo en `asigDSIC/ETSINF/dew/lab01/`

No debes modificarlo ni renombrarlo. Tus materiales se han de adaptarse a él.

A continuación se muestra la parte significativa del cuerpo del documento:

dew1 - Referencia al material entregado

Esta página `dew1.html` **NO** debe modificarse **NI** renombrarse. Son vuestros documentos los que deben ajustarse a esta página.

Supone la existencia de 2 directorios (sketchy y superhero) para diferenciar las 2 versiones de `ej1.html`

1. [p2.html](#) (procedente de `p2MAL.html`)
2. [ej1Fondamento.html](#) (fuente de Google Fondamento)
3. [ej1Montserrat.html](#) (fuente de Google Montserrat)
4. [ej1.html](#) (estilo Bootstrap+Sketchy)
5. [ej1.html](#) (estilo Bootstrap+Superhero)
6. [casiformToCheckDEW.html](#) (intento de formulario con comprobación)
7. [mi formToCheckDEW.html](#) (formulario con comprobación)

```
<body>
<h1>dew1 - Referencia al material entregado</h1>
<p>Esta página dew1.html NO debe modificarse
NI renombrarse. Son vuestros documentos los que deben ajustarse a
esta página.</p>
<p>Supone la existencia de 2 directorios (sketchy y superhero) para diferenciar las
2 versiones de ej1.html</p>
<ol>
  <li><a href="p2.html">p2.html</a> (procedente de p2MAL.html)</li>
  <li><a href="ej1Fondamento.html">ej1Fondamento.html</a> (fuente de Google
Fondamento)</li>
  <li><a href="ej1Montserrat.html">ej1Montserrat.html</a> (fuente de Google
Montserrat)</li>
  <li><a href="sketchy/ej1.html">ej1.html</a> (estilo Bootstrap+Sketchy)</li>
  <li><a href="superhero/ej1.html">ej1.html</a> (estilo Bootstrap+Superhero)</li>
  <li><a href="casiformToCheckDEW.html">casiformToCheckDEW.html</a> (intento de
formulario con comprobación)</li>
  <li><a href="mi formToCheckDEW.html">mi formToCheckDEW.html</a> (formulario con
comprobación)</li>
</ol>
</body>
</html>
```

13 BIBLIOGRAFÍA Y REFERENCIAS RÁPIDAS

En la guía docente se referencia una parte muy importante de la bibliografía aplicable a este tema. En general, dado un tópico, pueden encontrarse tanto libros enciclopédicos especializados como webs completas. Con el nivel de detalle que podemos dedicar a cada tópico, estas obras parecen excesivas.

Suele resultar cómodo emplear fuentes centralizadas de información como www.mcclibre.org/consultar/htmlcss, de Bartolomé Sintés, o bien www.w3schools.com, lugar inagotable con informaciones relacionadas con la web. Adicionalmente quiero citar librosweb.es como una alternativa, actualmente sin actividad, en español.

TÓPICO	REFERENCIA OFICIAL	TUTORIAL
HTML5	http://www.w3.org/TR/html/	
CSS	http://www.w3.org/Style/CSS/	http://librosweb.es/css/
JavaScript	http://www.ecmascript.org/	http://librosweb.es/javascript/
Bootstrap	http://getbootstrap.com/	http://librosweb.es/bootstrap_3/
Bootswatch	http://bootswatch.com/	
Eclipse (Oxygen)	http://www.eclipse.org/	http://help.eclipse.org/oxygen/index.jsp
Google web fonts	http://www.google.com/fonts	https://design.google/library/google-fonts/

ⁱ El hecho de que devuelva un documento XHTML es irrelevante