

Ejercicio 1: Mi primera Aplicación Web

Acerca del ejercicio

En este ejercicio aprenderemos a codificar nuestros primeros fragmentos de código HTML, CSS y PHP, pasando por el proceso de desplegar la aplicación en un servidor Apache.

Crearemos tres páginas:

- *index.html*: Es la página principal y lo único que contiene son los enlaces necesarios para acceder a las otras dos.
- *colores.html*: Es una página a la que aplicaremos distintas hojas de estilo para ver cómo va cambiando.
- *saludos.php*: Es un script en PHP que muestra por pantalla tantos saludos como reciba como parámetro.

Estas tres páginas deberán desplegarse en el servidor Apache que hay instalado en los laboratorios.

Primeros pasos: Apache y XAMPP

Como veremos con más detalle en temas posteriores, HTTPd Apache¹ es el servidor Web más popular en la actualidad. Es un proyecto de código libre con distribuciones para la mayoría de los sistemas operativos.

En su configuración más básica, Apache únicamente es capaz de servir páginas web estáticas (documentos HTML, documentos CSS, archivos binarios, etc.), pero la mayoría de instalaciones de Apache se configuran con uno o más *plugins* capaces de ejecutar scripts dinámicos para generar las respuestas a las peticiones recibidas.

De todos estos *plugins*, uno de los más comunes es PHP (también de código libre). Y cuando usamos PHP, lo más normal es que lo usemos para comunicarnos con una base de datos MySQL.

Servidor LAMP: En el mundo de las Aplicaciones Web, especialmente de código libre, es común usar el acrónimo LAMP para referirse a un servidor Linux, que tiene instalado Apache, MySQL y PHP.

Una de las formas más rápidas de preparar un entorno para trabajar con aplicaciones Web sería instalando el programa XAMPP², que ofrece un entorno preconfigurado con Apache, MySQL, PHP y Perl, para su instalación en cualquier sistema operativo³.

¹ <http://httpd.apache.org/>

² <http://www.apachefriends.org/index.html>

³ Si vas a instalar XAMPP en tu casa, lee el anexo que se encuentra al final del enunciado.

Los laboratorios de la Facultad tienen XAMPP preinstalado tanto en Linux como en Windows. También lo puedes instalar en tu ordenador y, en el caso de sistemas Windows, existe una distribución portable que no requiere instalación y se puede mantener en una llave USB.

Uso de XAMPP en Windows:

- Para arrancar el servidor, debes ir a [Inicio >>> Todos los programas >>> Programación >>> Programación Web >>> Arranque Apache].
- Abre un navegador y ve a la página <http://localhost> para comprobar que el servicio haya arrancado correctamente.
- La carpeta de trabajo para colocar tus archivos es:
`c:\software\programacion\www\xampp\htdocs`

Uso de XAMPP en Linux:

- Para arrancar el servidor, debes ir a [Aplicaciones >>> FdI-Laboratorios >>> Programación >>> Programación Web >>> Servicio Apache].
- Abre un navegador y ve a la página <http://localhost> para comprobar que el servicio haya arrancado correctamente.
- La carpeta de trabajo para colocar tus archivos es:
`/opt/xampp/htdocs`

Una vez encontrado el directorio de trabajo, puedes empezar a realizar los siguientes apartados.

Parte I: Página de entrada: index.html

Abre cualquier editor de texto (te recomendamos *Sublime Text* o *Notepad++*) y prepara un documento HTML con el aspecto de la figura 1.

- La página tiene un título de primer nivel (<h1>) y tres párrafos (<p>).
- El primer enlace apunta al documento `colores.html`
- El resto de enlaces enlace apuntan respectivamente a `saludos.php?num=10`, `saludos.php?num=42`, etc.

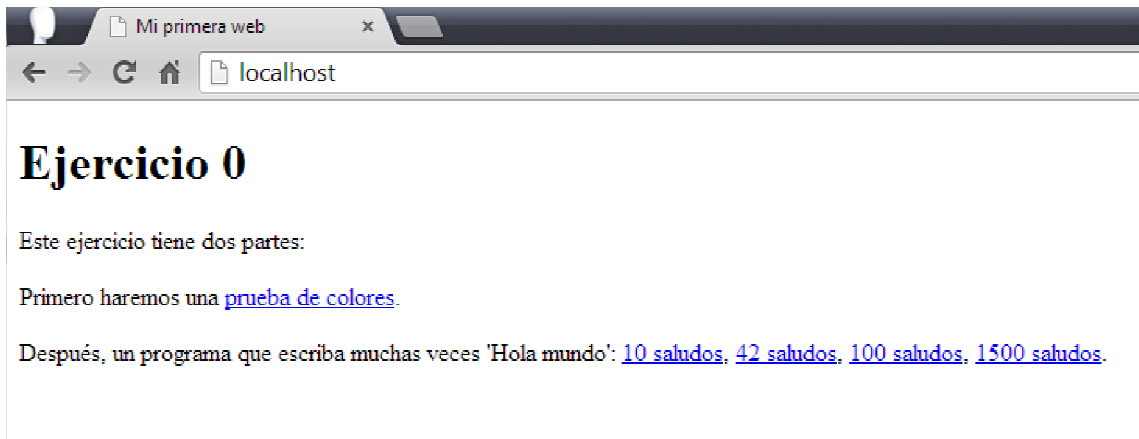
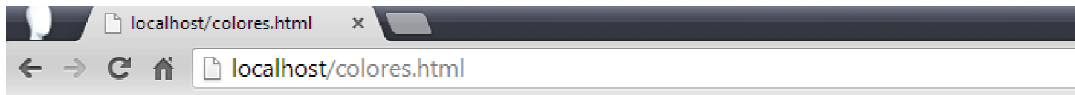


Fig. 1: Página inicial (index.html)

Parte II: Uso de CSS

En esta segunda página vamos a crear una página y cambiar su aspecto usando distintas hojas de estilo (CSS).

Empieza por crear una página web con el aspecto de la figura 2:



Página sencilla para decorar con estilos

Este es el primer párrafo, que no tiene nada de especial.

Esto es un título de segundo nivel

Que tiene dentro un párrafo con *un fragmento enfatizado*.

Esto es un título de tercer nivel

El tercer párrafo tampoco tiene nada de especial

Fig. 2: Página inicial (colores.html)

Esta página debe tener una hoja de estilos (default.css) para tener el mismo formato. Observa que todos los textos están en un tipo de letra *Sans Serif*, y que los fragmentos enfatizados () están en negrita.

Una vez que tengas la página, crea tres nuevas hojas de estilo para modificar el aspecto de la página:

- Estilo en blanco sobre verde (verde.css)
 - Color de la fuente: #FFFFFF
 - Color del fondo: #559955
 - Fuentes *Sans Serif*
- Estilo en marrón y grande (marron.css)
 - Color de la fuente: #FFCC00
 - Color del fondo: #5C1F00
 - Fuentes *Sans Serif*
 - Tamaño párrafo: 22px
 - Tamaño H1: 54px
 - Tamaño H2: 48px
 - Tamaño H3: 36px
- Estilo de consola (consola.css)
 - Color de la fuente: #00CC00
 - Color del fondo: #000000
 - Fuentes *monospace*
 - Tamaño para todo: 18px

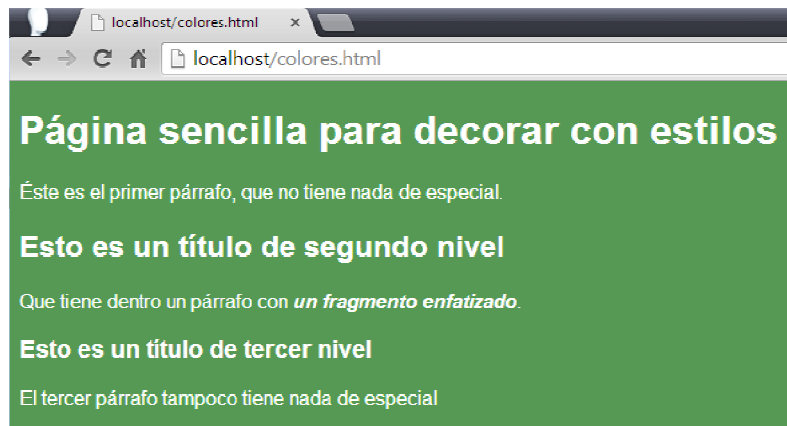


Fig. 3: Combinación en verde

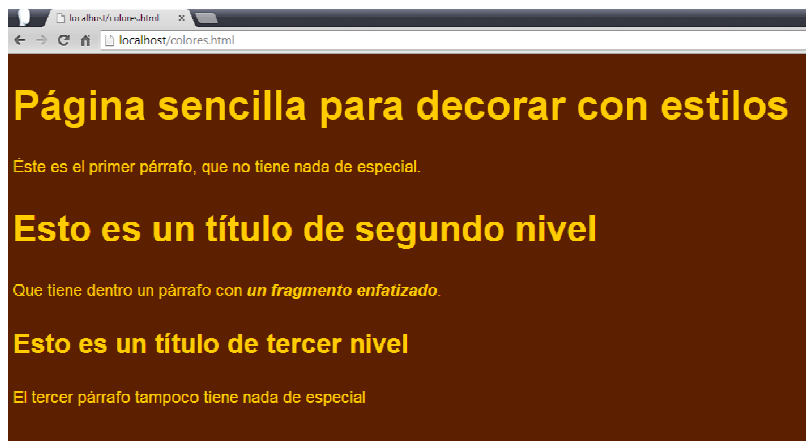


Fig. 4: Combinación en marrón con textos grandes

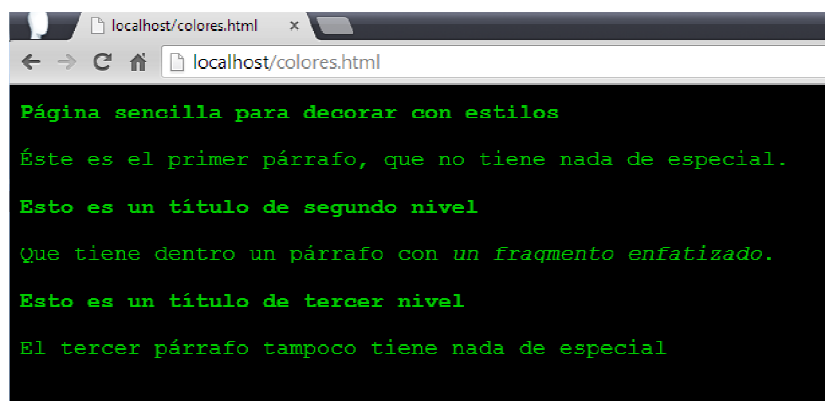
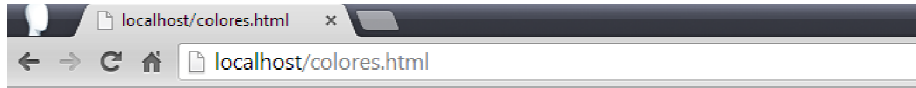


Fig. 5: Combinación en formato consola

Parte III: Uso de JavaScript

Modifica la página `colores.html` y añade tres botones que permitan activar las CSS sin recargar la página. Para ello, seguiremos los siguientes pasos:

- Añadir tres botones (`<button>`) en la parte inferior que permitan cambiar la CSS.
- Escribir una función en JavaScript que cambie la CSS.
- Vincular los botones con la nueva función.



Página sencilla para decorar con estilos

Éste es el primer párrafo, que no tiene nada de especial.

Esto es un título de segundo nivel

Que tiene dentro un párrafo con *un fragmento enfatizado*.

Esto es un título de tercer nivel

El tercer párrafo tampoco tiene nada de especial

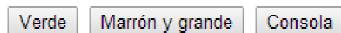


Fig. 6: Combinación en formato consola

Para conseguir que la CSS cambie dinámicamente, vamos a retocar el código de forma similar al ejemplo de JavaScript del Tema 0.

Primero, en la referencia a la hoja de estilos, añadimos una ID que nos facilite el acceso:

```
<link id="estilo" rel="stylesheet" type="text/css" href="default.css">
```

En los botones, pondremos una llamada a la función `cambiarCSS` con un parámetro que indique la nueva CSS:

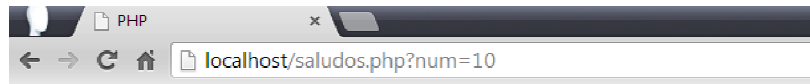
```
<button onclick="cambiarCSS('verde.css')">
```

Finalmente, escribe la función JavaScript en la que, en lugar de cambiar el `innerHTML`, vamos a cambiar el atributo `href`:

```
function cambiarCSS(nuevo){
  document.getElementById('estilo').setAttribute('href', nuevo);
}
```

Parte IV: Código en el servidor

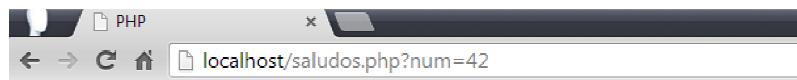
La última parte consiste en mejorar el programa `saludos.php` que vimos en el tema 0. En lugar de siempre mostrar el número de saludos tal cual, vamos a hacer que si recibe un número en concreto (el 42) no se muestren los saludos sino una frase distinta.



Me has pedido que te salude 10 veces.

0 - ¡Hola Mundo!
1 - ¡Hola Mundo!
2 - ¡Hola Mundo!
3 - ¡Hola Mundo!
4 - ¡Hola Mundo!
5 - ¡Hola Mundo!
6 - ¡Hola Mundo!
7 - ¡Hola Mundo!
8 - ¡Hola Mundo!
9 - ¡Hola Mundo!

Fig. 7: Llamada al script de saludos con num=10.



Me has pedido que te salude 42 veces.

En realidad, esa es la respuesta a la pregunta final sobre la vida, el universo y todo lo demás.

Fig. 8: Llamada al script de saludos con num=42.

Parte V: Simulación HTTP

Vamos a simular el código HTTP que envía nuestro navegador al servidor para obtener el resultado de las Figuras 7 y 8. Para ello necesitarás un cliente de telnet. Encontrarás uno en las consolas de todos los sistemas operativos. En Windows puedes utilizar PuTTY.

Para realizar las pruebas debes conectar a **localhost** en el puerto **80**. Una vez conectado probaremos tanto el método GET como POST.

1. Método GET

Simplemente escribe la siguiente línea:

```
GET /sw/ej01/saludos.php?num=3 HTTP/1.0
```

Comprueba el resultado. Ahora repite la conexión pero modificando la línea anterior para obtener el resultado de la Figura 8.

2. Método POST

Ahora queremos repetir el ejercicio pero simulando una petición POST donde los datos se envían junto a los datos de la petición en vez de a través de la URL.

Primero debes cambiar el código PHP de saludos.php para que obtenga el parámetro “num” del contenido POST en vez de GET. Simplemente cambia el array **\$_GET** por **\$_POST**. La petición HTTP a realizar es la siguiente (ten en cuenta el salto de línea entre la cabecera y el contenido):

```
POST /sw/ej01/saludos.php HTTP/1.0
Content-type: application/x-www-form-urlencoded
Content-length: 5
```

```
num=3
```

Para poder introducir el contenido del mensaje POST (num=3) no podrás escribir el texto directamente en la consola de telnet. En vez de eso copia y pega las líneas anteriores.

Entrega del trabajo

Una vez terminado el ejercicio, comprime todos los archivos (HTML, PHP, CSS) y copia el resultado de la parte V en un fichero de texto. Además añade otro archivo de texto con los nombres de los alumnos que entregan el trabajo (o con tu nombre si has hecho el ejercicio en solitario).

Este archivo comprimido se debe subir a la entrega habilitada en el campus virtual antes del final de la clase.

Si el ejercicio está hecho en pareja, **ambos miembros deben subir el ejercicio**.

ANEXO: Instalando XAMPP en mi ordenador

Acerca de XAMPP, la configuración y la seguridad

XAMPP es un entorno de desarrollo muy interesante y fácil de configurar, pero un servidor Web es una herramienta compleja y potencialmente peligrosa, pues expone puertos en el servidor y podría ser una vía de entrada para atacantes externos.

Si instalas XAMPP, es importante que cuides los aspectos de seguridad y que entiendas bien lo que está pasando.

Configuración de puertos

Al abrir XAMPP, verás que tienes la opción de lanzar los distintos servicios. Cada uno de estos servicios intentará abrir un puerto en tu ordenador para que pueda recibir conexiones desde equipos externos a través de Internet.

En particular, los servicios de XAMPP intentarán abrir los siguientes puertos:

- Apache: 80 (HTTP) y 443 (HTTPS)
- MySQL: 3306
- FileZilla: 21 (FTP)

Para que estos puertos sean accesibles desde Internet, tendrás que permitir el acceso a través del firewall de tu sistema operativo (o incluso de la configuración NAT de tu servidor).

XAMPP y Skype: Por defecto, Skype ocupa el puerto 80 cada vez que se lanza, por lo que es incompatible con XAMPP. Si lanzas XAMPP con Skype encendido, el servidor Apache fallará al arrancar. Puedes aprender más sobre esto aquí:

<https://www.youtube.com/watch?v=wl6gyfY7QKw>

Seguridad

Abrir puertos en tu ordenador, en general, supone exponerte a vulnerabilidades. Ten en cuenta que XAMPP está pensado para que sea muy fácil ponerse a desarrollar, primando sencillez sobre seguridad. Por tanto, no es recomendable usarlo para un servidor permanente y accesible en Internet. Si vas a tener XAMPP activado en tu ordenador, dedica un rato a aprender sobre cómo mejorar su seguridad:

- <http://robsnotebook.com/xampp-security-hardening>
- <https://ajbalmon.wordpress.com/2008/07/01/configurando-xampp/>