

Práctica 1.
-------------

**Nombre de la práctica:** Instalación de un Entorno de desarrollo Web.

**Objetivo:** Identificar, Instalar y Configurar las herramientas necesarias para un ambiente de desarrollo Web.

En esta práctica se identifican, instalan y configuran las tecnologías más utilizadas en el desarrollo de aplicaciones Web, es decir, los editores, lenguajes, servidores y demás aplicaciones necesarias para programar y/o generar documentos HTML5. También se recomendarán herramientas CASE que faciliten el trabajo del estudiante y que aporten velocidad y claridad en el desarrollo Web. Finalmente se ejemplificará de manera sencilla el desarrollo de una aplicación Web “de juguete”, es decir, con fines meramente demostrativos pero que intentarán dejar claro el papel de cada una de las tecnologías abordadas.

**Introducción:** La arquitectura de una aplicación se refiere a cómo ésta será construida. La elección de la arquitectura con la cual se construye una aplicación está determinada por el usuario final, es decir, la persona o dispositivo que harán uso del software que se va a desarrollar. Se pueden distinguir dos tipos de arquitectura: Una que es centralizada, es decir, que tanto datos como procedimientos residen o son administrados por la aplicación que se está desarrollando y otra que es distribuida y en la cual la administración de los datos está repartida entre varias aplicaciones, por lo que muchas de las tareas de gestión de datos ya están previamente construidas.

Derivado de la propagación de redes de computadoras en el mundo, las aplicaciones distribuidas evolucionaron hasta llegar a una arquitectura denominada cliente-servidor, la cual separa datos de procedimientos además de incorporarse un controlador que funge como enlace entre la aplicación cliente y la aplicación servidor.

Con el crecimiento de Internet y con el apoyo de otras tecnologías como el protocolo HTTP, el cual permite la transferencia de los datos entre el cliente y el servidor, surgen las aplicaciones Web. Una aplicación Web es una aplicación cliente-servidor en la cual el cliente es un navegador Web. Las aplicaciones Web utilizan documentos escritos en HTML para presentar los datos al cliente final, aunque también incorporan tecnologías como JavaScript y CSS. Los documentos HTML son administrados y/o generados por otro programa llamado Servidor de Aplicaciones o Servidor Web. El Servidor de Aplicaciones se encarga de integrar varias tecnologías para procesar las peticiones del Cliente y generar las respuestas en documentos HTML. Finalmente estos documentos son formateados, validados y desplegados por el navegador.

Considerando lo anterior, además de considerar principios básicos de ingeniería de software, a continuación se listan y describen de manera general un grupo de herramientas ampliamente probadas para desarrollar aplicaciones Web y que en su conjunto conformarán nuestro entorno de desarrollo Web durante todo el curso.

- ❖ Servidor de Aplicaciones. Se trata de una aplicación que se encarga de gestionar la lógica de negocio y el acceso a los datos de una aplicación Web. Por tal razón también se le conoce como Servidor Web. En esencia podemos configurar nuestro servidor web por dos razones fundamentales: para crear un entorno de desarrollo web o para publicar documentos y aplicaciones web. En esta práctica abordaremos el primero de los casos.
- ❖ Editor de Diagramas. Una aplicación Web requiere del uso de las mejores prácticas recomendadas por la ingeniería de software y parte de ello es el modelado y el diseño a través de distintos diagramas como el Conceptual, el de Casos de Uso, el de Clases, Modelo EER, etc.
- ❖ Herramienta de Maquetado. Estas aplicaciones nos permiten ir definiendo los formularios con los que el usuario final va a interactuar.
- ❖ SGBD. Los Sistemas Gestores de Base de Datos son las aplicaciones que administrarán los datos detrás de bambalinas.
- ❖ Lenguajes de programación. Generalmente se utilizan distintos lenguajes de programación para crear una aplicación web, dependiendo si se va a interactuar con el usuario final o con los datos o con el SGBD.
- ❖ Software para control de versiones. Las aplicaciones web, por su naturaleza, se desarrollan en diferentes etapas, con distintas tecnologías y en distintas etapas. La administración del desarrollo en sí se convierte en una actividad crítica.

#### Material y equipo necesario:

- a. Se requiere de un equipo de cómputo con un sistema operativo instalado y con los permisos necesarios para llevar a cabo todas las instalaciones, configuraciones y pruebas correspondientes.
- b. Se requiere conexión a internet o bien contar con el software previamente descargado para su instalación.
- c. Se requiere de al menos un navegador web.
- d. Tiempo estimado 4 horas.

#### Metodología:

1. Descargar e instalar un editor de texto. A continuación se listan algunos que recomendamos:
  - a. Notepad++ (Windows), <https://notepad-plus-plus.org/>
  - b. Gedit (Windows, MacOS), <http://www.gedit.org>
  - c. Atom (Windows), <https://atom.io/>
  - d. Sublime Text (Windows), <https://www.sublimetext.com/>
2. Descargar e instalar el editor de diagramas DIA el cual es multiplataforma y que soporta UML. <http://dia-installer.de/>
3. Descargar e instalar el Servidor de Aplicaciones. Existen diferentes formas de hacer esto, pero para esta práctica obviaremos algunos pasos y descargaremos la suite completa de aplicaciones que vamos a necesitar, es decir: Apache como servidor web, MySQL (o

MariaDB) como gestor de base de datos y PHP o Python como lenguaje de programación del lado del servidor. Por sus iniciales y anteponiendo la inicial del sistema operativo de nuestro gusto podemos recurrir a WAMP, LAMP o MAMP o la genérica XAMP. Para poder instalar alguna de estas suites debemos ingresar a los sitios oficiales de las mismas:

- a. <http://www.wampserver.com/en/>
  - b. <https://www.apachefriends.org/es/index.html>
  - c. <http://www.ampps.com/downloads>
4. Descargar e instalar una aplicación cliente para MySQL/MariaDB. En nuestro caso utilizaremos dos distintos durante este curso:
- a. SQLyog, <https://github.com/webeyog/sqlyog-community/wiki/Downloads>
  - b. WorkBench, <https://dev.mysql.com/downloads/workbench/>
5. Utilizando un editor de texto crear un archivo llamado index.html tal y como se muestra en el listado1, para este ejercicio la ubicación del mismo no es importante. Aunque la extensión ya es propia de un documento html evidentemente no se trata de tal. Solamente es un archivo de texto plano.
6. Ahora busquemos abrir este documento con un navegador web, esta operación se puede hacer de múltiples formas aunque un doble clic en el explorador de archivos será suficiente. Tómame algunos minutos para escribir en tus conclusiones qué crees lo que ha pasado al abrir este documento con el navegador web.
7. Insertemos ahora si etiquetas html en el index.html que definen de manera básica, aunque no completa, la estructura de un documento html5. Las etiquetas son insertadas

```
1 Tema 1. Introducción
2 1.1 Evolución de las aplicaciones web
3 1.2 Arquitectura de las aplicaciones web
4 1.3 Tecnologías para el desarrollo de aplicaciones web
5 1.4 Planificación de aplicaciones web
6 Tema 2. Lenguajes de marcado
7 2.1 Introducción
8 2.2 Estructura de un documento web
9 2.3 Elementos básicos
10 2.4 Formularios
11 2.5 Lenguaje de presentación
12 2.6 Selectores
13 2.7 Modelo de Caja
14 Tema 3. Programación del lado del cliente
15 3.1 Introducción al lenguaje
16 3.2 Manejo de frameworks
17 3.3 Estructuras de control
18 3.4 Manipulación de objetos
19 Tema 4. Programación del lado del servidor
20 4.1 Introducción al lenguaje
21 4.2 Estructuras de control
22 4.3 Tratamiento de formularios
23 4.4 Manejo de objetos
24 4.5 Creación de clases
25 4.6 Acceso a datos
26 Tema 5. Cómputo en la nube
27 5.1 Conceptos generales
28 5.2 Tipos de servicios en la nube
29 5.3 Patrones de diseño
30 5.4 Estándares en servicios
31 5.5 Plataformas tecnológicas
32 5.6 Seguridad e interoperabilidad
33
```

Listado 1 Contenido del index.html correspondiente a la primera práctica.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Programación Web</title>
5 </head>
6 <body>
7   Tema 1. Introducción
8   1.1 Evolución de las aplicaciones web
9   1.2 Arquitectura de las aplicaciones web
10  1.3 Tecnologías para el desarrollo de a
11  1.4 Planificación de aplicaciones web
12  Tema 2. Lenguajes de marcado
13  2.1 Introducción
```

Listado 2 Encabezado básico de un documento html5.

al inicio y agregadas al final del documento tal y como se muestra en los listados 2 y 3. La etiqueta **<body>** marca el inicio de los contenidos del documento html, es decir, los elementos que el navegador web desplegará. Las etiquetas que están fuera del “cuerpo” son informativas para el navegador pero no se desplegarán. Se puede observar también en este ejemplo que todas las etiquetas son compuestas, es decir, que se requiere de una etiqueta de cierre que corresponda a otra que abre o marca el inicio estructural del documento html.

Almacene el archivo y visualice el documento con el navegador. Enriquezca sus conclusiones con el resultado obtenido.

8. Aunque este documento de ejemplo es un listado de temas y subtemas y html cuenta con etiquetas apropiadas para este fin, utilizaremos sin embargo otras destinadas a jerarquizar elementos dentro de un documento. Se trata de las etiquetas “H”, las cuales siguen siendo etiquetas estructurales que le indican al navegador la importancia dentro del documento que tiene el contenido que se esté referenciando. Las de mayor jerarquía son **<h1>**, **<h2>**, etc., y las de menor jerarquía son **<h6>** y **<h7>**. Es importante entender que aunque cada navegador le da un estilo o apariencia distinta a cada uno de estos elementos el objetivo de los mismos no es ese, sino informarle a los motores de búsqueda la jerarquía de los contenidos dentro de un documento. Encerrar todos los temas y subtemas del ejemplo utilizando dos etiquetas “H” distintas tal y como se muestra en el listado 4. Visualizar el resultado en un navegador y anotar sus conclusiones.

```

36      5.4 Estándares en servicios
37      5.5 Plataformas tecnológicas
38      5.6 Seguridad e interoperabilidad
39    </body>
40  </html>
41
42

```

Listado 4 Etiquetas que marcan el cierre del documento html5.

```

6    <body>
7      <h1>Tema 1. Introducción</h1>
8      <h2>1.1 Evolución de las aplicaciones web</h2>
9      <h2>1.2 Arquitectura de las aplicaciones web</h2>
10     <h2>1.3 Tecnologías para el desarrollo de aplicaciones web</h2>
11     <h2>1.4 Planificación de aplicaciones web</h2>
12     <h1>Tema 2. Lenguajes de marcado</h1>
13     <h2>2.1 Introducción</h2>
14     <h2>2.2 Estructura de un documento web</h2>
15     <h2>2.3 Elementos básicos</h2>
16     <h2>2.4 Formularios</h2>

```

Listado 3 Etiquetas estructurales “H”.

9. Finalmente vamos a agregar elementos vinculatorios a nuestro documento html utilizando la etiqueta **<a>** (su nombre viene del inglés “*anchor*” o “*ancla*” en español). Esto nos permitirá vincular cada subtema con un correspondiente contenido, que de hecho es la esencia de los documentos basados en hipertexto. En el listado 5 se ilustra el uso de esta etiqueta.

```

<h1>Tema 1. Introducción</h1>
<a href="evolucion.html">
  <h2>1.1 Evolución de las aplicaciones web</h2>
</a>

```

Listado 5 Uso de la etiqueta <a> para vincular contenidos.

10. Para que funcione el vínculo definido en el paso anterior será necesario crear un documento adicional llamado **evolucion.html**, en este documento el estudiante colocará un contenido asociado al tema que se esté desarrollando. Por ejemplo para este primer tema, “*Evolución de las aplicaciones Web*” se incluirá texto plano (sin formato) separado en párrafos. Sin embargo ya hemos visto que un documento web no respeta necesariamente la disposición del texto que hayamos usado en el editor. No reconoce espacios, ni tabulaciones, ni saltos de líneas. Por lo tanto tendremos que utilizar las etiquetas correspondientes para poder distribuir el contenido en nuestro documento web de la manera más atractiva para el lector. Una idea simple es distribuir el contenido en párrafos, esto en html se logra con la etiqueta compuesta **<p>...</p>**, colocando entre ellas cada uno de los párrafos. Un ejemplo para este documento puede ser el mostrado en el listado 6. No importa la distribución que hagamos del contenido en el editor de textos al final serán las etiquetas las que determinen su distribución en el navegador web. Observar como al final se agregó un

anclaje para regresar al menú inicial. Transcriban el contenido del listado 6 (o crear el propio) y ver los resultados en el navegador. Agregar lo observado a las conclusiones.

```
<!DOCTYPE html>
<html lang="es">
  <head> <title>Programación Web</title> </head>
  <body>
    <h1>1.1 Evolución de las aplicaciones web</h1>
    <p>
      Una Aplicación Web o también llamada web app, es una aplicación
      Cliente-Servidor en la cual el cliente o la interfaz de usuario
      la proporciona un navegador web. Dentro de las aplicaciones web
      podemos incluir el webmail, wikis, mensajería instantánea y
      prácticamente todo lo que funciona a través de un navegador.
    </p>
    <p>
      En un principio, cada página web no era sino un documento estático
      en donde el usuario podía navegar entre documentos gracias a las
      bondades del hiper texto. El precio a pagar es que con cada clic
      se tenía que visitar al servidor para acceder a un nuevo recurso.
    </p>
    <p>
      En 1995, Netscape introduce un lenguaje de script del lado del
      cliente llamado JavaScript con el cual el documento podía hacerse
      dinámico sin necesidad de acceder al servidor. Es decir, el usuario
      podía interactuar con el documento localmente.
    </p>
    <p>
      En 1996, Macromedia introduce Flash, el cual permitió agregar a los
      navegadores animaciones embebidas en las páginas web.
    </p>
    <a href="index.html"><h3>Regresar</h3></a>
  </body>
</html>
```

Listado 6 Contenido del archivo "evolucion.html".

Extra clase.

Terminar esta práctica generando los contenidos de todos los subtemas antes señalados. Respetar el uso de las etiquetas que aquí se han visto. El estudiante puede echar un vistazo y utilizar también para este ejercicio otras etiquetas diseñadas para la manipulación de texto como:

- |            |          |           |
|------------|----------|-----------|
| ○ <br>     | ○ <b>    | ○ <time>  |
| ○ <span>   | ○ <cite> | ○ <small> |
| ○ <strong> | ○ <data> | ○ <code>  |
| ○ <em>     | ○ <kbd>  | ○ <samp>  |
| ○ <sub>    | ○ <i>    | ○ <img>   |
| ○ <sup>    | ○ <mark> |           |

Conclusiones:

Referencias: