



I. TEMA: CONFIGURACIÓN DE UN SERVIDOR WEB EN EL SISTEMA OPERATIVO LINUX

II. OBJETIVOS

Al finalizar la presente práctica el estudiante:

1. Implementa un servidor Web para una red local
2. Diseña páginas web utilizando HTML5, CSS3 y bases de datos.
3. Implementa un servidor web con dominios virtuales.
4. Integra el servidor web y el servidor de nombres (DNS)

III. PRE-REQUISITOS

Para realizar satisfactoriamente la presente práctica de laboratorio es necesario:

1. Conocimientos previos del protocolo HTTP
2. Manejo básico del sistema operativo Linux
3. Conocimientos básicos de HTML.
4. Conocimientos básicos de CSS.
5. Conocimientos básicos de bases de datos.

IV. MATERIALES Y EQUIPOS.

Los materiales que utilizaremos en los trabajos de laboratorio son:

1. Sistema operativo Linux – distribución Debian.
2. Manuales de configuración del servidor web Apache versión 2.x
3. Manuales de programación HTML 5, CSS, Bases de datos.



V. MARCO TEÓRICO.

HTTP

HTTP es un protocolo de la pila de protocolos TCP/IP, cuyo propósito es ofrecer servicios de transferencia de hipertexto, que pueden ser accedidos desde clientes conocidos como navegadores web.

HTML – HyperText Markup Language

HTML es un lenguaje de etiquetas para la construcción de documentos web.

La última versión de HTML es la versión 5, que incorpora mejoras sustanciales respecto de las versiones anteriores tales como funciones específicas para manejo de gráficos, audio y video embebido.

Algunos de los tags que define HTML 5 son:

<header> </header> - Especifica el encabezado de un documento o una sección.

<nav> </nav> - representa una sección de una página que vincula a otras páginas o a partes de la página: una sección con enlaces de navegación.

<article> </article> - Representa un artículo que es un contenido independiente o autocontenido en el sitio.

<section> </section> - Representa una sección genérica de un documento. También una agrupación temática de contenido que puede incluir un encabezado. Puede incluirse dentro de secciones, artículos o bandas laterales.

<aside> </aside> - Representa una sección de una página que no contiene el contenido principal de la página sino contenido indirectamente relacionado al contenido principal.

<hgroup> (i.e., <h1> a <h6>) - Representa el encabezado de una sección, que consiste de todos los elementos hijo <h1> a <h6> del elemento hgroup. El elemento indica sub encabezados o subtítulos.

<a href> - La etiqueta de enlace o de hiperenlace. Esta hace al texto “hiper” de modo que cuando se hace click sobre este podemos cargar otra página o activar algún programa en JavaScript.

Para una documentación completa de las etiquetas HTML5 visite el sitio web del estándar: <http://www.w3.org/TR/html5/>.



CSS – Cascading Style Sheet

CCS permite definir el formato y estilo de presentación de páginas web escritas en HTML.

CSS define la forma de representación de las páginas web, utilizando reglas que se almacenan en un archivo diferente al archivo HTML que define el contenido web, de esa forma, se separa el contenido web de su forma de presentación.

Por cuanto el estilo se independiza del contenido, CSS permite utilizar las hojas de estilo como plantillas para mantener consistente el diseño de las páginas web.

Una regla CSS comprende los siguientes elementos:

1. Selector
2. Declaración
3. Propiedad
4. Valores

Un ejemplo de una regla CSS se describe a continuación:

```
body {  
    font-style : normal  
    color      : green  
}
```

En esta regla se indica que la sección `body` utilizará una fuente normal de color verde.

Para una descripción mas amplia y actual, se recomienda visitar el sitio web de World Wide Web Consortium (W3C), donde se establece y define los estándares relacionados con las tecnologías web: <https://www.w3.org/Style/CSS/>.

Javascript

Javascript es un lenguaje de programación, que puede utilizarse para ejecutar programas en un navegador web (no solamente).

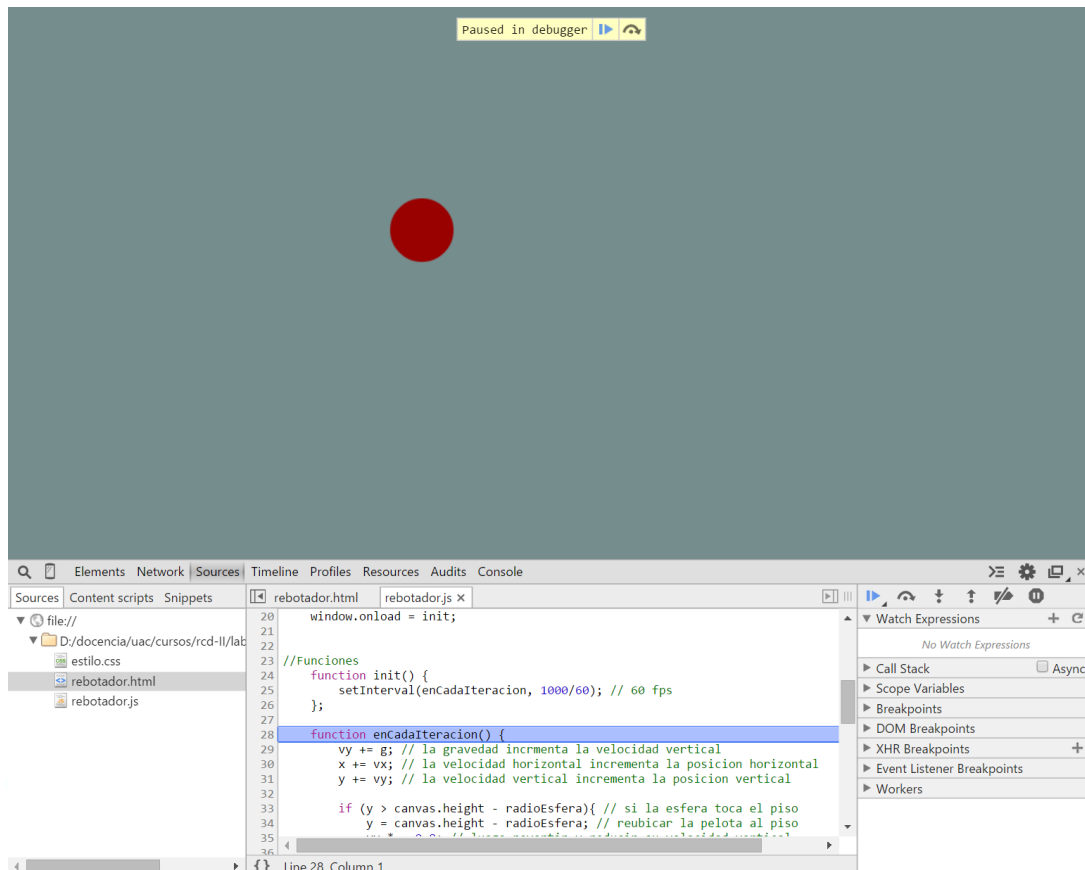
Mientras que HTML es un lenguaje que especifica el contenido de una página web y CSS define la forma de presentación de las páginas web, Javascript permite realizar acciones en base a un algoritmo, dotando a las páginas web de la capacidad de poder realizar acciones interactivas.

Un programa en Javascript puede embeberse en un documento HTML para su ejecución en el cliente. El código se descarga junto con la página web y el archivo de estilos. Luego el navegador web interpreta el código Javascript localmente.



INTEGRACION DE HTML, CSS Y JAVASCRIPT

La página web de la imagen muestra el resultado de la ejecución del programa `rebotador.js`, en Javascript, que dibuja una esfera que rebota sobre el piso y que va perdiendo altura en cada rebote. El código se embebe (incrusta) en el archivo `rebotador.html`, que utiliza el archivo `estilo.css` para definir el color de fondo de la pantalla:



El código fuente en Javascript de `rebotador.js` es:

```
//Nombre      : Rebotador
//Proposito   : Mostrar esfera que rebota
//Autor       : Edwin Carrasco (adaptacion de codigo de ejemplo
               de autoria no determinada)
//FCreacion   : 21/10/2021

//Variables
var canvas = document.getElementById('canvas');
var context = canvas.getContext('2d');
var radioEsfera = 30;
var color = "#dd0000";
var g = 0.1; // aceleración por la gravedad
var x = 50; // posicion horizontal inicial
var y = 50; // posicion vertical inicial
```



UNIVERSIDAD ANDINA DEL CUSCO

ADMINISTRACIÓN DE SERVICIOS DE REDES

GUÍA DE LABORATORIO

ECP 5 de 22

```
var vx = 2; // velocidad horizontal inicial
var vy = 0; // velocidad vertical inicial

//Accion
window.onload = init;

//Funciones
function init() {
    setInterval(enCadaIteracion, 1000/60); // 60 fps
};

function enCadaIteracion() {
    vy += g; //la gravedad incrementa la velocidad vertical
    x += vx; //la velocidad horizontal incrementa la posicion
            horizontal
    y += vy; //la velocidad vertical incrementa la posicion
            vertical

    //si la esfera toca el piso
    if (y > canvas.height - radioEsfera){
        //reposicionar la pelota en el piso
        y = canvas.height - radioEsfera;
        //luego revertir y reducir su velocidad vertical
        vy *= -0.8;
    }

    //si la esfera sale del canvas
    if (x > canvas.width + radioEsfera){
        //volver por extremo izquierdo (repetir)
        x = -radioEsfera;
    }

    dibujarEsfera(); //dibujar la esfera
};

function dibujarEsfera() {
    with (context) {
        clearRect(0, 0, canvas.width, canvas.height);
        fillStyle = color;
        beginPath();
        arc(x, y, radioEsfera, 0, 2*Math.PI, true);
        closePath();
        fill();
    };
};
```

El código del archivo CSS estilo.css es:

```
*{
    background: #ABCCCC;
}
```



El código HTML de la página web rebotador.html:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Rebotador</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <canvas id="canvas" width="1000" height="500"></canvas>
    <script src="rebotador.js"></script>
  </body>
</html>
```

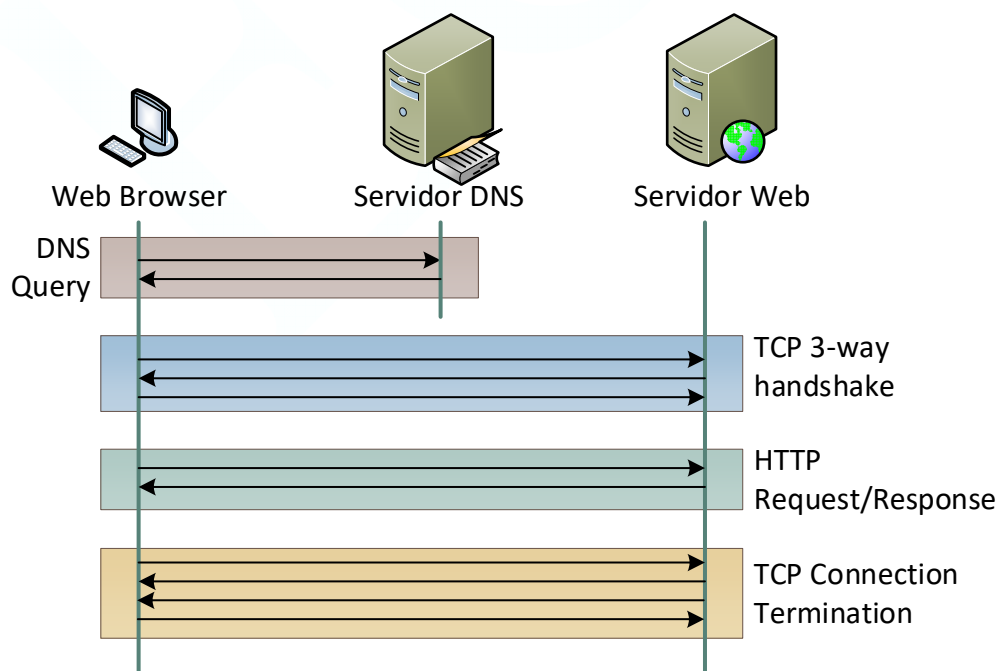
FLUJO DE INTERACCION ENTRE UN CLIENTE Y UN SERVIDOR WEB.

Como en todo proceso de comunicación con servicios de red accesibles a través de un URL, el primer paso consiste en consultar al DNS por la dirección IP del servidor web.

El DNS responde enviando al cliente web la dirección IP del servidor web.

Con la dirección IP del servidor web, el cliente web establece una conexión TCP a través de la cual enviará las peticiones HTTP y recibirá las respuestas HTTP.

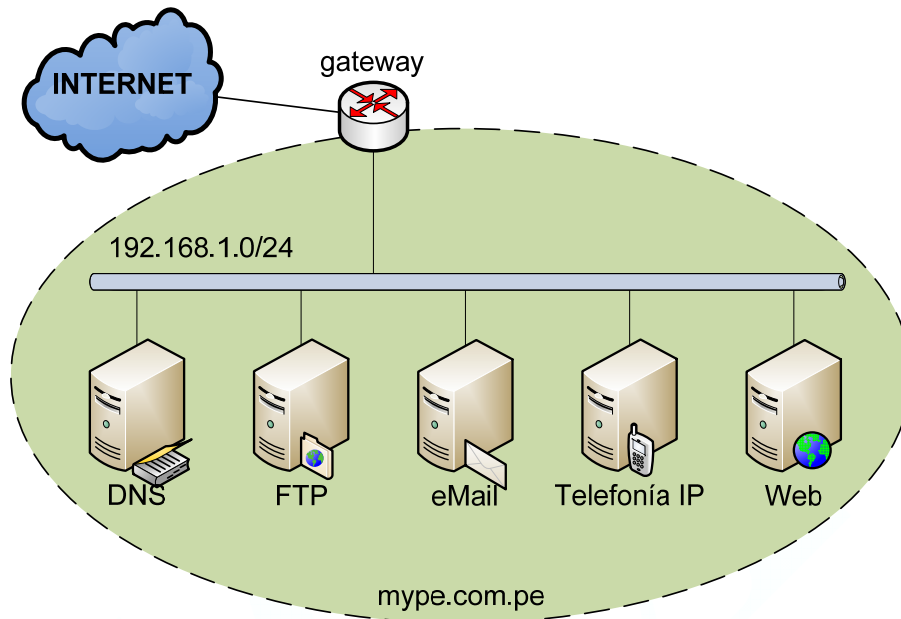
Culminada la recuperación de objetos web, el cliente cierra la conexión TCP con el servidor web.





VI. PRACTICAS DE LABORATORIO.

Para realizar el presente trabajo de laboratorio vamos a tomar como red de ejemplo una compuesta por servidores y hosts como se muestra en el siguiente gráfico, cuyo nombre de dominio es mype.com.pe y cuya dirección de red es 192.168.1.0/255.255.255.0



Las especificaciones adicionales de los equipos se muestran en el siguiente cuadro:

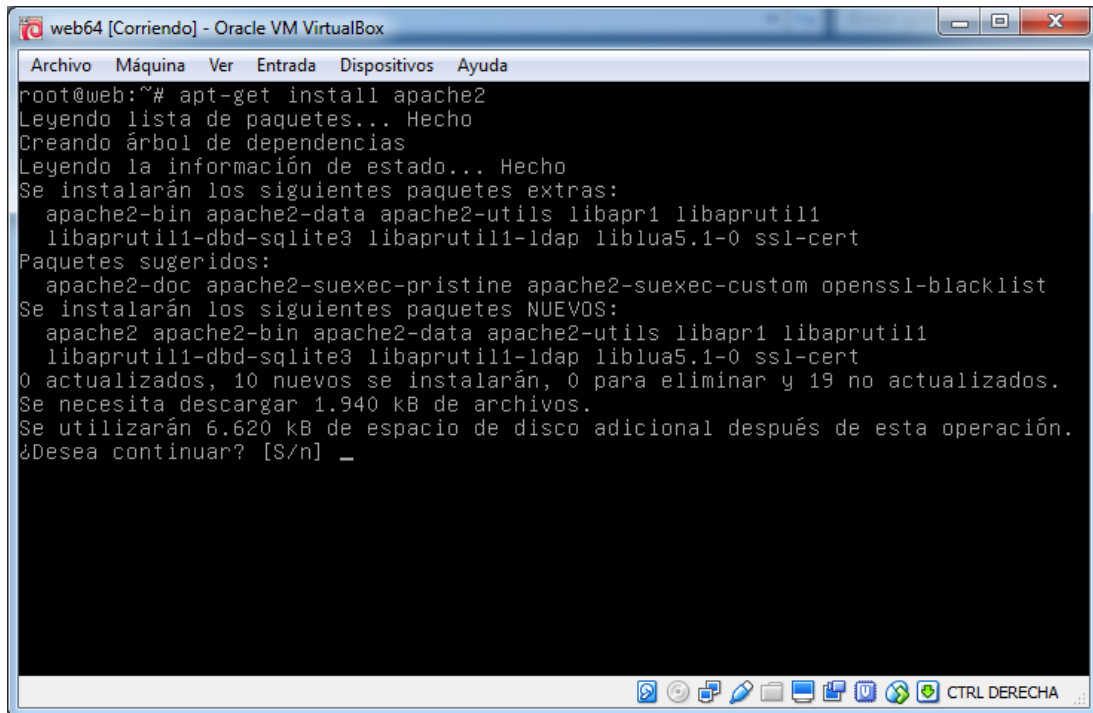
SERVICIO	Nombre de equipo	Dirección IP
DNS	dns	192.168.1.254/24
FTP	ftp	192.168.1.253/24
Web	web	192.168.1.252/24
eMail	correo	192.168.1.251/24
Telefonía IP	centralita	192.168.1.250/24
Router	gateway	192.168.1.1/24



GESTIÓN Y CONFIGURACIÓN DEL SERVIDOR APACHE

El proceso de configuración del servidor web comienza instalando el servidor web seleccionado. En nuestro caso, el servidor Apache versión 2. Para ello ejecutamos en la consola del equipo Linux la orden:

```
apt-get install apache2
```



```
web64 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
root@web:~# apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom openssl-blacklist
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 19 no actualizados.
Se necesita descargar 1.940 kB de archivos.
Se utilizarán 6.620 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

Para detener el servicio web utilizamos el comando

```
/etc/init.d/apache2 stop
```

Para iniciar el servicio web utilizamos el comando

```
/etc/init.d/apache2 start
```

Para reiniciar el servicio utilizamos el commando

```
/etc/init.d/apache2 restart
```

La configuración por defecto del servidor web Apache se establece en el archivo `/etc/apache2/apache2.conf`.

Para el presente trabajo, no se hará ningún cambio en la instalación por defecto.

Los cambios se harán en la carpeta que almacena los archivos del sitio web que se publicará.



PARÁMETROS DE CONFIGURACIÓN DEL SERVIDOR WEB APACHE

Se comentan, a continuación, algunos de los parámetros más importantes para la operación del servidor web Apache

ServerRoot – La raíz del árbol de directorio bajo el cual se mantienen los archivos de configuración, error y log.

PidFile – El archivo en el cual el servidor debe grabar su PID cuando se inicia.

Timeout – El número de segundos antes que se venza el envío y recepción de información.

KeepAlive – Permite o no conexiones persistentes (más de un requerimiento por conexión).

MaxKeepAliveRequests – El número máximo de requerimientos que se permite durante una conexión persistente.

KeepAliveTimeout – Número de segundos que se debe esperar el siguiente requerimiento del mismo cliente a través de la misma conexión.

AccessFileName – el nombre del archivo que debe revisarse en cada directorio para determinar directivas de configuración adicionales

DefaultType – El tipo MIME por defecto que utilizara el servidor en caso que no pueda determinar el tipo de un archivo.

HostNameLookups – grabar en el archivo log los nombres de los clientes o solo sus direcciones IP.

ErrorLog – La ubicación del archivo de registro de errores.

LogLevel – Controla el número de mensajes guardados en el archivo de registro errores (debug, info, notice, warn, error, crit, alert, emerg).

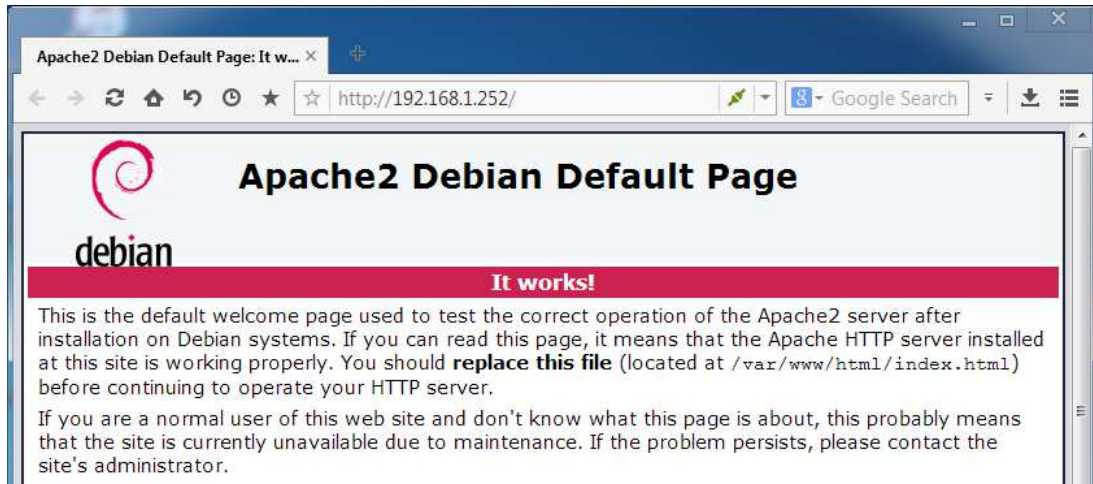
CONEXIÓN DESDE UN CLIENTE WEB

Para verificar la funcionalidad del servidor web instalado, nos conectamos desde un navegador web en un equipo diferente al servidor.

En el navegador web escribimos la URL <http://192.168.1.252> y, si la instalación fue correcta, deberíamos visualizar la página por defecto del servidor web Apache.

Inicialmente probamos la conexión utilizando la dirección IP de nuestro servidor (192.168.1.252), por lo que no es necesario hacer referencia al servidor DNS.

La página web mostrada (index.html) esta almacenada en la carpeta /var/www/html, que es el directorio raíz del servidor web Apache en Debian.



Como se puede ver, el servidor web se encuentra operando.

Para construir un sitio web que reemplace al creado por defecto, vamos a crear una página web propia.



CREACIÓN DE UNA PAGINA WEB PERSONALIZADA

Para crear una página web podemos utilizar cualquier editor de texto y escribir las páginas utilizando código HTML. Sin embargo, si deseamos construir y mantener un sitio web, es mejor utilizar herramientas que faciliten las tareas de diseño y mantenimiento. Algunas herramientas que nos permiten realizar estas tareas son:

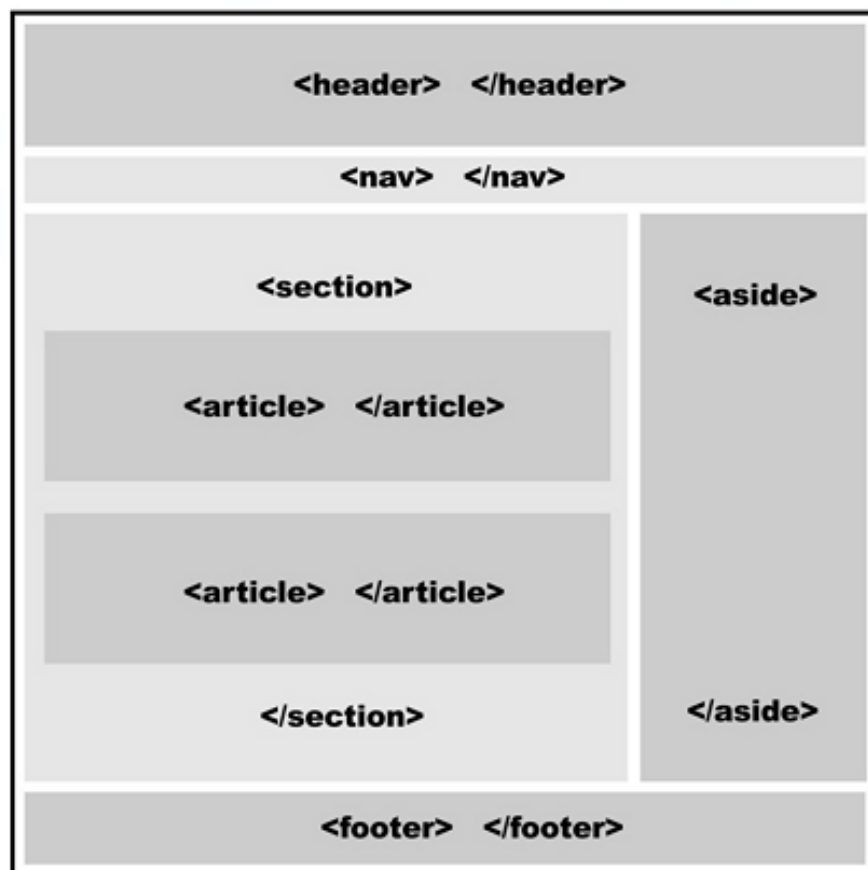
- Nvu
- Kompozer
- Dreamweaver

De estas, las primeras son herramientas de software libre. Pueden ser adecuadas para proyectos pequeños, pero para proyectos grandes presentan algunas limitaciones, como por ejemplo, la administración de sitios web.

Dreamweaver es una aplicación propietaria, mucho más profesional que las anteriores y permite la administración de sitios web, la conexión y mantenimiento de los sitios web mediante FTP, etc.

Para la presente guía, utilizaremos un editor de texto sencillo: notepad++.

El diseño de nuestra página web seguirá la estructura indicada en el siguiente gráfico:





El código HTML5 para la página index.html es:

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="iso-8859-1"/>
  <title> Portal de mype.com.pe </title>
  <link rel="stylesheet" href="estilo.css">
</head>

<body>
<div id="agrupar">
  <header id="cabecera">
    <h1> Bienvenidos a mype.com.pe</h1>
  </header>

  <nav id="menu">
    <ul>
      <li>Principal</li>
      <li>Servicios</li>
      <li>Productos</li>
      <li>Comunidad</li>
      <li>Contactenos</li>
    </ul>
  </nav>

  <section id="seccion">
    <article>
      mype es una organizacion que presta servicios y
      soluciones basadas en tecnologias de información y
      comunicaciones (TIC) adaptadas a los requerimientos de
      las organizaciones.
      Contamos con una amplia cartera de clientes y una vasta
      experiencia, puesta a su servicio.
    </article>

    <article>
      Prestamos asesoria en:
      <ul>
        <li id="lista">Proyectos de redes de computadores
          bajo especificaciones de cableado estructurado</li>
        <li id="lista">Proyectos de telefonía IP</li>
        <li id="lista">Proyectos de redes inalámbricas
          rurales</li>
        <li id="lista">Proyectos de desarrollo e
          implementación de sistemas de información</li>
        <li id="lista">Proyectos de capacitación en TIC</li>
      </ul>
    </article>
  </section>

  <aside id="columna">
    <figure>
```



```

</figure>
<blockquote>Historias de exito</blockquote>
<blockquote>Nos recomiendan</blockquote>
</aside>
<footer id="pie">
  <small>mype.com.pe - Derechos reservados &copy;
2015</small>
</footer>
</div>
</body>
</html>
```

Para definir la forma en que debe representarse la página, definimos en el archivo `estilo.css` los estilos para cada elemento de la página web.

El archivo `estilo.css` debe contener:

```
* {
  margin: 0px;
  padding: 0px;
}

h1 {
  font: bold 20px verdana, sans-serif;
}

h2 {
  font: bold 14px verdana, sans-serif;
}

header, section, footer, aside, nav, article, figure,
figcaption, hgroup{
  display: block;
}

body {
  text-align: center;
}

#agrupar {
  width: 960px;
  margin: 15px auto;
  text-align: left;
}

#cabecera {
  background: #CFEBB9;
  border: 1px solid #999999;
  padding: 20px;
}
```



UNIVERSIDAD ANDINA DEL CUSCO

ADMINISTRACIÓN DE SERVICIOS DE REDES

GUÍA DE LABORATORIO

ECP 14 de 22

```
#menu {
  background: #CCCCCC;
  border: 1px solid #999999;
  padding: 5px 15px;
}

#menu li {
  display: inline-block;
  list-style: none;
  padding: 5px;
  font: bold 14px verdana, sans-serif;
}

#seccion {
  float: left;
  width: 660px;
  margin: 20px;
}

#lista {
  margin: 20px;
}

#columna {
  float: left;
  width: 220px;
  margin: 20px 5px;
  padding: 5px;
  background: #ABCCCC;
}

#pie {
  clear: both;
  text-align: center;
  padding: 20px;
  border-top: 2px solid #999999;
}

article {
  background: #99FBCC;
  border: 1px solid #999999;
  padding: 20px;
  margin-bottom: 15px;
}

figcaption {
  font: italic 14px verdana, sans-serif;
}
```

Con un navegador web cualquiera podemos verificar la forma en que se visualizará nuestra página:



UNIVERSIDAD ANDINA DEL CUSCO

ADMINISTRACIÓN DE SERVICIOS DE REDES

GUÍA DE LABORATORIO

ECP 15 de 22



Una vez terminado el diseño de la página web, esta debe subirse al servidor web.

En Debian, la página por defecto del servidor Web se almacena en la carpeta /var/www/html. Los archivos que se utilizaron para crear la página web (archivo HTML, imágenes y otros) se deben copiar a la carpeta indicada

```
root@web:/var/www/html# ls -la
total 44
drwxr-xr-x 2 root root 4096 ago 15 23:39 .
drwxr-xr-x 3 root root 4096 ago 13 23:44 ..
-rwxr-xr-x 1 root root 26930 ago 15 23:39 core.jpg
-rwxr-xr-x 1 root root 1042 ago 15 23:39 estilo.css
-rwxr-xr-x 1 root root 1606 ago 15 23:39 index.html
```

PRUEBA DE ACCESO DESDE CLIENTE WEB

Si no hubo problemas al diseñar la página ni al subir la información; al acceder al sitio web desde un navegador web, el resultado será:

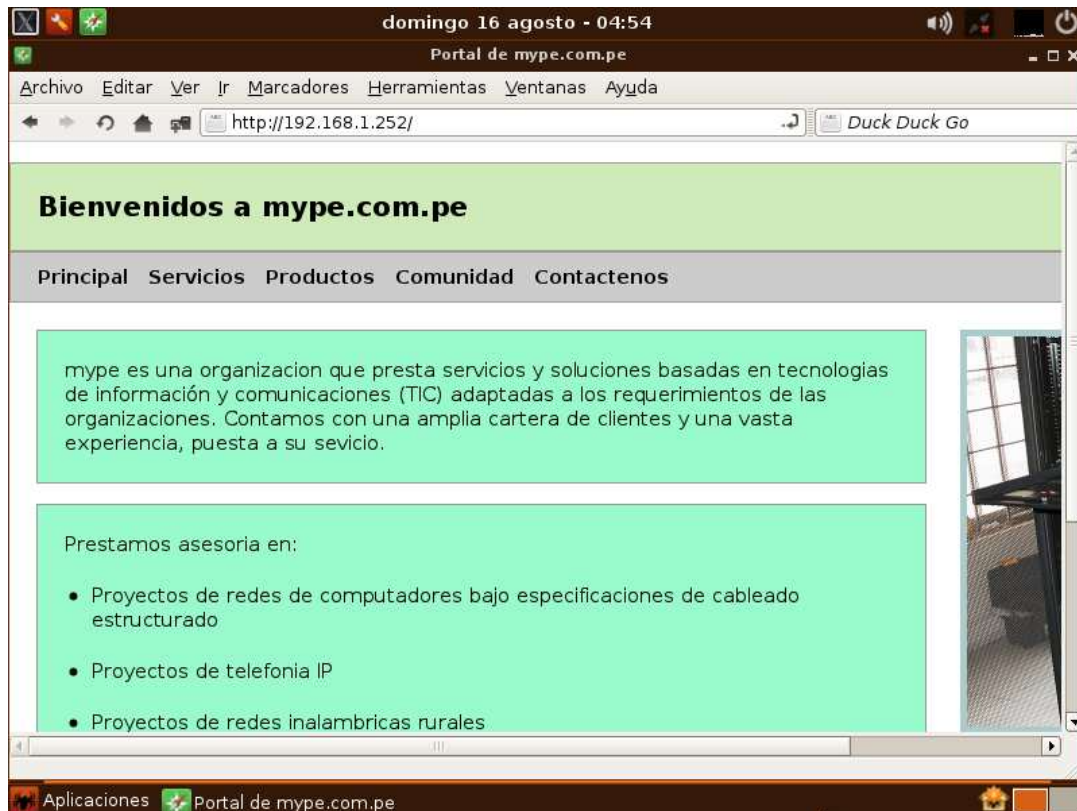


UNIVERSIDAD ANDINA DEL CUSCO

ADMINISTRACIÓN DE SERVICIOS DE REDES

GUÍA DE LABORATORIO

ECP 16 de 22



Se demuestra que el servidor web es funcional y el contenido web se representa correctamente de acuerdo al diseño establecido en la hoja de estilos.

El siguiente paso será permitir que los usuarios puedan acceder al servicio web mediante el nombre www.mype.com.pe. Para lograr este propósito, debemos agregar el nombre y la dirección IP del servidor web en los registros del servidor DNS.



UNIVERSIDAD ANDINA DEL CUSCO

ADMINISTRACIÓN DE SERVICIOS DE REDES

GUÍA DE LABORATORIO

ECP 17 de 22

ACTUALIZACIÓN DEL SERVICIO DNS

Si modificamos la configuración del servidor DNS para que resuelva el nombre de nuestro servidor web, la conexión debe poder hacerse mediante el nombre del mismo (www.mype.com.pe). Para conseguir esto, seguimos los siguientes pasos:

En el servidor DNS:

Editamos `/etc/bind/db.mype`, agregando el registro:

```
www A 192.168.1.252
```

```
GNU nano 2.2.6 Fichero: db.mype
$TTL 1D
@      IN SOA dns.mype.com.pe. hostmaster.mype.com.pe. (
        201507251      ;serial
        8H            ;refresh
        4H            ;retry
        4W            ;expiracion
        1D            ;minimo
        )

        NS dns.mype.com.pe.

localhost      A      127.0.0.1
dns            A      192.168.1.254
www            A      192.168.1.252
```

Editamos `/etc/bind/db.rev.192.168.1` agregando el registro:

```
252 PTR www.mype.com.pe.
```

```
GNU nano 2.2.6 Fichero: db.rev.192.168.1
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 1D
@      IN SOA dns.mype.com.pe. hostmaster.mype.com.pe (
        201507251;      serial
        8H;          refresh
        4H;          retry
        4W;          expiracion
        1D;          minimo
        )

        NS      dns.mype.com.pe.

254    PTR      dns.mype.com.pe.
252    PTR      www.mype.com.pe.
```

Reiniciamos el servicio DNS

```
root@dns:/etc/bind# rndc reload
server reload successful
root@dns:/etc/bind# _
```



UNIVERSIDAD ANDINA DEL CUSCO

ADMINISTRACIÓN DE SERVICIOS DE REDES

GUÍA DE LABORATORIO

ECP 18 de 22

En el equipo cliente Linux, modificamos el archivo /etc/resolv.conf:

```
sakura
root@slitaz:/# echo 'nameserver 192.168.1.254' > /etc/resolv.conf
root@slitaz:/#
```

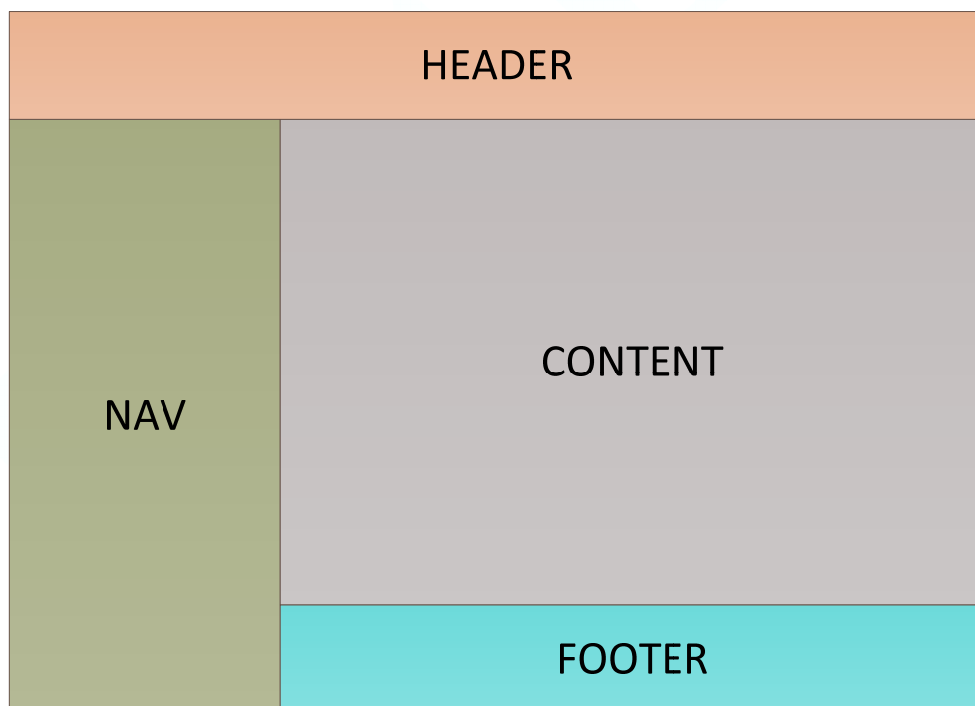
Volvemos a ingresar a la página web de mype.com.pe mediante el nombre del servicio www.mype.com.pe. Si la configuración fue correcta, debemos obtener un resultado similar al mostrado en la siguiente figura:





VII. TRABAJOS DE LABORATORIO.

1. Investigue el concepto de dominios virtuales y agregue dos dominios virtuales al dominio mype.com.pe. Implemente una página acorde para cada dominio y verifique que todos los dominios puedan ser accedidos mediante sus nombres de dominio. Por ejemplo, a través de www.mype.com.pe se debe acceder a la página principal de mype.com.pe; a través de www.finanzas.mype.com.pe se debe acceder a la página web de la unidad de finanzas de mype.com.pe y a través de www.logistica.mype.com.pe, se debe acceder a la página web de la unidad de logística de mype.com.pe.
2. Para el diseño de las páginas web de los dominios de finanzas y logística siga las siguientes especificaciones:
 - a) Las páginas web utilizan HTML estático con la plantilla CSS mostrada en el siguiente gráfico:



- b) Las páginas de finanzas y prospectiva deben permitir listar los nombres y apellidos de sus respectivos trabajadores.

Para este ejercicio se debe crear una base de datos y recuperar los datos utilizando el lenguaje de programación de su preferencia. La base de datos tendrá la siguiente estructura:

tTrabajadores(CodTrabajador, ApPaterno, Nombre, Area).



UNIVERSIDAD ANDINA DEL CUSCO
ADMINISTRACIÓN DE SERVICIOS DE REDES
GUÍA DE LABORATORIO

ECP 20 de 22

3. Documente, utilizando Wireshark, una sesión entre un dominio virtual de mype y un cliente en Windows o Linux, incluyendo la interacción con el servidor DNS del dominio mype.com.pe.

Se debe mostrar de manera explícita la resolución de nombre, el establecimiento de la conexión TCP, la comunicación HTTP (REQUEST/RESPONSE), la transferencia de los objetos HTML, estilos CSS, imágenes y consultas y respuestas entre el cliente y el servidor web y el cierre de la conexión TCP.

Los segmentos y mensajes intercambiados deben mostrarse de acuerdo al formato estándar de los protocolos correspondientes.

La documentación debe imprimirse en una hoja A3 o A2 para su calificación.



UNIVERSIDAD ANDINA DEL CUSCO
ADMINISTRACIÓN DE SERVICIOS DE REDES
GUÍA DE LABORATORIO

ECP 21 de 22

VIII. EVALUACIÓN

La evaluación de las actividades realizadas en la presente guía de práctica se hará en función de la siguiente tabla:

ACTIVIDAD	PROCEDIMENTAL		
	SESIÓN 01	SESIÓN 02	SESIÓN 03
Resolución del ejercicio propuesto 01	--	12	--
Resolución del ejercicio propuesto 02	--	08	--
Resolución del ejercicio propuesto 03	--	--	20
TOTAL	--	20	20



IX. REFERENCIAS

1. Adobe. “Software de Diseño de Sitios Web | Adobe Dreamweaver”
<http://www.adobe.com/es/products/dreamweaver/>
2. Asipa Dolamu. “Cascading Style Sheets (CSS) for Dummies”.
<https://hackernoon.com/cascading-style-sheets-css-for-dummies>
3. Athiban S. “HTML Semantic Elements”. <https://senthurathiban.medium.com/html-semantic-elements-e08524f727b>
4. Carrasco E. “Configuración de un Servidor Web en el Sistema Operativo Linux”.
<https://campus.uandina.edu.pe/mod/resource/view.php?id=1072836>
5. Both D. “How To Configure Multiple Websites With Apache Web Server”
<https://opensource.com/article/18/3/configuring-multiple-web-sites-apache>
6. Comer “Internetworking with TCP/IP. Volumen I”. Ed. Prentice Hall 4ed
7. Kompozer. “Kompozer Easy Web Authoring”. <http://kompozer.net/>
8. Lin Y., Hwang R., Baker F. “Computer Networks. An Open Source Approach”. Mc Graw Hill 2012.
9. NVU.”NVU”. <http://www.nvu.com/>
10. Shah S., Soyinka W. “Manual de administración de Linux” cuarta edición, McGraw Hill 2007.
11. The Apache Software Foundation. “Documentación del Servidor HTTP Apache 2.4”.
<http://httpd.apache.org/docs/current/>
12. Warrenne Blane. “Securing Your Apache 2 Server with SSL”.
<https://www.sitepoint.com/securing-apache-2-server-ssl/>
13. Wshito. “Bola que rebota en AsciiDoc”.
<http://diary.wshito.com/comp/js/canvas/bouncing-ball/>