

Práctica World Wide Web

tp2 redes practico

Problema 1

Sin utilizar un editor especializado en html, y diseñe y muestre el código HTML de cada una de las siguientes páginas para que se vean en un cliente Web de la siguiente manera

Página del Núcleo Básico Obligatorio:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Núcleo Básico Obligatorio</title>
</head>
<body>
  
  <h1>Tecnicatura en Programación Informática</h1>
  <h2>Materias del núcleo básico obligatorio:</h2>
  <ul>
    <li>Introducción a la Programación</li>
    <li>Organización de Computadoras</li>
    <li>Matemática I</li>
    <li>Programación con Objetos I</li>
    <li>Bases de Datos</li>
    <li>Estructuras de Datos</li>
    <li>Programación con Objetos II</li>
  </ul>
  <a href="index.html">Volver a la página principal de la UNQ</a>
</body>
</html>
```

Página del Núcleo Avanzado Obligatorio:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Núcleo Avanzado Obligatorio</title>
</head>
<body>
  
  <h1>Tecnicatura en Programación Informática</h1>
  <h2>Materias del núcleo avanzado obligatorio:</h2>
  <p>Las materias del núcleo avanzado obligatorio completan la formación obligatoria del estudiante. Para todas las materias incluidas en la siguiente tabla, el régimen de cursado es cuatrimestral, y la modalidad es presencial.</p>
```

```

<table>
  <tr>
    <th>Materia</th>
    <th>Horas semanales</th>
    <th>Carga horaria</th>
    <th>Créditos</th>
  </tr>
  <tr>
    <td>Redes de Computadoras</td>
    <td>6</td>
    <td>108</td>
    <td>12</td>
  </tr>
  <tr>
    <td>Sistemas Operativos</td>
    <td>6</td>
    <td>108</td>
    <td>12</td>
  </tr>
  <tr>
    <td>Programación Concurrente</td>
    <td>4</td>
    <td>72</td>
    <td>8</td>
  </tr>
  <tr>
    <td>Matemática II</td>
    <td>4</td>
    <td>72</td>
    <td>8</td>
  </tr>
</table>
<a href="index.html">Volver a la página principal de la UNQ</a>
</body>
</html>

```

Página del Núcleo Complementario:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Núcleo Complementario</title>
</head>
<body>
  
  <h1>Tecnicatura en Programación Informática</h1>
  <h2>Asignaturas del núcleo complementario:</h2>

```

<p>Las materias del núcleo complementario permiten orientar al estudiante hacia un perfil determinado dentro del universo amplio y cambiante de los proyectos de desarrollo de software.</p>

<button onclick="location.href='search.html'">Búsqueda</button>

Volver a la página principal de la UNQ

</body>

</html>

Problema 2

El protocolo HTTP permite hacer distintos tipos de pedidos para recursos de un determinado dominio. Escriba los requerimientos

HTTP 1.1 que permitan obtener los siguientes pedidos al sitio web del departamento de computación (www.dc.uba.ar):

- El recurso /
- Encabezado del recurso /tac
- El recurso /logo.jpg si no fue modificado desde una determinada fecha

Crearemos un archivo de texto con las instrucciones dadas para los requerimientos HTTP

1. Obtener el recurso principal ("/")

Para solicitar el recurso principal del sitio, que usualmente corresponde a la página de inicio, se utiliza el método `GET`. El requerimiento se vería de la siguiente manera:

GET / HTTP/1.1

Host: www.dc.uba.ar

2. Obtener el encabezado del recurso "/tac"

Para solicitar solo los encabezados de un recurso específico (en este caso, `/tac`), sin descargar todo el contenido del recurso, se utiliza el método `HEAD`. Este método es útil para obtener metadatos como el tipo de contenido o la última fecha de modificación, sin necesidad de descargar todo el recurso. El requerimiento sería:

HEAD /tac HTTP/1.1

Host: www.dc.uba.ar

3. Solicitar el recurso "/logo.jpg" solo si no ha sido modificado desde una fecha específica

Para solicitar un recurso pero indicando que solo se desea recibirlo si ha sido modificado después de una determinada fecha, se utiliza el método `GET` junto con el encabezado `If-Modified-Since`. Este encabezado permite realizar una solicitud condicional basada en la fecha de modificación del recurso. Por ejemplo, si quieres solicitar el recurso `/logo.jpg` solo si no ha sido modificado desde el 1 de enero de 2023, el requerimiento se vería así:

GET /logo.jpg HTTP/1.1

Host: www.dc.uba.ar
If-Modified-Since: Sat, 01 Jan 2023 00:00:00 GMT

Problema 3

Explique las diferencias entre estas dos secuencias de comandos realizadas desde un servidor cualquiera en Internet:

telnet www.inta.gob.ar 80

GET / HTTP/1.1

Host: www.inta.gob.ar

telnet www.inta.gob.ar 80

GET / HTTP/1.1

Host: www.mercosurt.org.ar

Las dos secuencias de comandos usan `telnet` para conectar al puerto 80 de un servidor web y enviar solicitudes HTTP, pero difieren en el encabezado `Host` que especifican:

1. La primera secuencia solicita el recurso principal de `www.inta.gob.ar`, conectándose a este servidor y usando el mismo en el encabezado `Host`. Esto es una solicitud estándar y correcta.
2. La segunda secuencia también se conecta a `www.inta.gob.ar`, pero solicita el recurso como si fuera de `www.mercosurt.org.ar` en el encabezado `Host`. Esto puede causar errores o comportamientos inesperados, ya que el servidor puede no estar configurado para responder por el dominio especificado en el encabezado `Host`.

La diferencia clave es que la primera secuencia es coherente en la conexión y el encabezado `Host`, mientras que la segunda mezcla la conexión a un servidor con un encabezado `Host` para otro dominio, lo cual generalmente no es una práctica correcta.

Problema 4

Una empresa decide instalar una plataforma de servicios web. Se espera que se conecten hasta 5 clientes simultáneamente.

1. ¿Cuántos servidores web son necesarios?
2. ¿Cuántas direcciones IP hacen falta?
3. ¿En cuántos puertos TCP diferentes deben estar siendo atendidos?
4. ¿Y si fueran 150.000.000 de clientes?

###

1. ¿Cuántos servidores web son necesarios?

Para manejar hasta 5 clientes simultáneamente, ****técnicamente solo se necesita 1 servidor web****. Los servidores web modernos están diseñados para manejar múltiples conexiones simultáneas, y la cantidad de clientes que un servidor puede manejar al mismo tiempo depende de su configuración, recursos disponibles (como CPU, memoria RAM), y la complejidad de las tareas que realiza. Para 5 clientes, un servidor adecuadamente configurado debería ser suficiente.

2. ¿Cuántas direcciones IP hacen falta?

Solo se necesita ****1 dirección IP pública**** para el servidor web, independientemente de la cantidad de clientes que se conecten simultáneamente. La dirección IP se usa para identificar al servidor en la red. Los clientes utilizan esta dirección IP para establecer la conexión con el servidor.

3. ¿En cuántos puertos TCP diferentes deben estar siendo atendidos?

Para servicios web estándar, generalmente solo se necesitan ****2 puertos TCP****: uno para HTTP (puerto 80) y otro para HTTPS (puerto 443, la versión segura de HTTP).

Independientemente de la cantidad de clientes, estos dos puertos pueden manejar todas las conexiones entrantes si solo se está sirviendo contenido web.

4. ¿Y si fueran 150.000.000 de clientes?

- ****Servidores web necesarios****: Con 150.000.000 de clientes, se necesitaría una infraestructura mucho más grande, probablemente compuesta por múltiples servidores web (o incluso cientos o miles, dependiendo de la carga específica), distribuidos potencialmente a través de un sistema de balanceo de carga o una red de entrega de contenido (CDN) para manejar eficientemente tal volumen de tráfico.

- ****Direcciones IP hacen falta****: A nivel de servidores web expuestos al internet, aún se podría iniciar con una sola dirección IP pública si se utiliza un balanceador de carga. Sin embargo, en la práctica, se usarían múltiples direcciones IP en un escenario de alta disponibilidad y escalabilidad.

- ****Puertos TCP diferentes****: A pesar del aumento masivo en el número de clientes, los servicios web estándar aún solo necesitarían estar atendidos en los mismos 2 puertos TCP (80 para HTTP y 443 para HTTPS). El uso eficiente de estos puertos se maneja a través de la infraestructura de red y servidor, como balanceadores de carga.

- ****Infraestructura adicional****: Manejar 150.000.000 de clientes simultáneamente requeriría también consideraciones adicionales como optimización de aplicaciones, sistemas de gestión de tráfico, seguridad reforzada, y posiblemente servicios en la nube para escalar dinámicamente según la demanda.

###

En resumen, mientras que la infraestructura necesaria escala significativamente con el número de clientes, los principios básicos de cómo se utilizan las direcciones IP y los puertos TCP permanecen consistentes. La complejidad adicional viene en la gestión y escalabilidad de la infraestructura para soportar la carga.

Problema 5

Interprete la salida del siguiente comando que ha sido realizado desde una PC cualquiera conectada a Internet:

Comando:

```
$ telnet www.inta.gov.ar 80
```

```
HEAD / HTTP/1.1
```

```
Host: www.inta.gov.ar
```

Salida:

```
HTTP/1.1 200 OK
```

```
Accept-Ranges: bytes
```

```
Date: Thu, 24 Mar 2011 23:15:43 GMT
```

```
Content-Length: 286
```

```
Content-Type: text/html
```

Last-Modified: Wed, 06 Oct 2010 12:46:39 GMT
ETag: "749338855465cb1:0"
Server: Microsoft-IIS/7.0
X-Powered-By: ASP.NET

La salida del comando `telnet` seguido de la solicitud `HEAD` hacia `www.inta.gov.ar` en el puerto 80 muestra la respuesta HTTP del servidor a dicha solicitud.

- ****HTTP/1.1 200 OK****: La respuesta comienza con el protocolo HTTP versión 1.1 seguido del código de estado 200, que indica que la solicitud ha sido exitosa y el servidor ha encontrado el recurso solicitado.
- ****Accept-Ranges: bytes****: Esta línea indica que el servidor acepta solicitudes de rango, lo que permite a los clientes descargar solo partes específicas de un recurso, en lugar de todo el recurso de una vez.
- ****Date: Thu, 24 Mar 2011 23:15:43 GMT****: Fecha y hora en la que el servidor procesó la solicitud, en formato GMT.
- ****Content-Length: 286****: El tamaño del recurso solicitado, en este caso, el tamaño del documento HTML principal es de 286 bytes.
- ****Content-Type: text/html****: El tipo de contenido del recurso solicitado es texto HTML, lo que significa que se trata de una página web.
- ****Last-Modified: Wed, 06 Oct 2010 12:46:39 GMT****: Fecha y hora de la última modificación del recurso solicitado.
- ****ETag: "749338855465cb1:0"*****: Un identificador único asignado al recurso solicitado. Puede ser utilizado por los clientes para determinar si el contenido ha cambiado desde la última vez que se accedió, lo que permite optimizar la caché.
- ****Server: Microsoft-IIS/7.0****: La versión del software del servidor web que está atendiendo la solicitud, en este caso, Microsoft Internet Information Services versión 7.0.
- ****X-Powered-By: ASP.NET****: Indica que el sitio web está construido o impulsado por ASP.NET, un marco de aplicación web desarrollado por Microsoft.

Problema 6

Al visualizar el contenido de una página web se bajan archivos de los siguientes servidores:

- server1: 3 archivos jpg y un html
- server2: 2 archivos gif
- server3: 1 archivo wav

¿Cuántas conexiones de nivel de transporte (TCP) se realizan si se utiliza HTTP/1.0? ¿Y se utiliza HTTP/1.1?

En HTTP/1.0, cada recurso (como una imagen o una página web) requiere abrir y cerrar una conexión TCP propia porque no reutiliza conexiones. Esto significa:

- **Server1** necesita 4 conexiones TCP (3 jpg + 1 html).
- **Server2** necesita 2 conexiones TCP (2 gif).
- **Server3** necesita 1 conexión TCP (1 wav).
- **Total en HTTP/1.0**: 7 conexiones TCP.

En HTTP/1.1, las conexiones TCP pueden reutilizarse gracias al mecanismo "keep-alive", reduciendo el número de conexiones necesarias. Así que, teóricamente:

- **Se necesita 1 conexión TCP por servidor**, independientemente del número de recursos.
- **Total en HTTP/1.1**: 3 conexiones TCP (1 por cada servidor).

Esto hace que HTTP/1.1 sea más eficiente en cuanto al uso de conexiones.

Problema 7

Un sitio web tiene una página inicial (index.html), que referencia a dos archivos jpg en dos servidores distintos.

¿Cuántas conexiones de nivel de transporte (TCP) se realizarán usando HTTP/1.1 cuando se visite dicho sitio?

Usando HTTP/1.1, el comportamiento por defecto permite la reutilización de conexiones TCP gracias al mecanismo "keep-alive", lo que significa que las conexiones se pueden mantener abiertas para múltiples solicitudes y respuestas. Sin embargo, la capacidad de reutilizar una conexión depende de que las solicitudes sean hacia el mismo servidor. Dado que los recursos (archivos jpg) están en servidores distintos, este comportamiento afecta cómo contamos las conexiones:

- **1 conexión TCP** para la página inicial ('index.html').
- **1 conexión TCP** para el primer archivo jpg (en un servidor distinto).
- **1 conexión TCP** para el segundo archivo jpg (en otro servidor distinto).

Total: 3 conexiones TCP serían realizadas usando HTTP/1.1 cuando se visite el sitio, una para cada recurso solicitado de servidores diferentes.

Problema 8

Suponga la siguiente página escrita en HTML que reside en el servidor www.unq.edu.ar:

```
<html>
<head>
<title></title>
</head>
<body>
<h1>hola mundo!!</h1>
<h2><a href="http://www.unq.edu.ar/"></a>
html hecho realmente fácil</h2>
```

```
<a href="http://mail.google.com"></a>  
</body>  
</html>
```

¿Cuántas conexiones de nivel de transporte (TCP) utiliza el navegador para transferir la totalidad de la información usando HTTP/1.0?

Usando HTTP/1.0, cada recurso requerido por la página web establece y cierra su propia conexión TCP, ya que HTTP/1.0 no soporta de forma nativa la reutilización de conexiones TCP. Esto implica:

1. ****1 conexión TCP**** para la página HTML principal (`index.html`).
2. Aunque el enlace en el `h2` lleva a `www.unq.edu.ar`, como es solo un enlace y no se solicita automáticamente con la carga de la página, no cuenta como una conexión adicional en este contexto.
3. ****1 conexión TCP**** para el archivo de imagen `logo_transp.gif` de `www.unq.edu.ar`.
4. ****1 conexión TCP**** para el archivo de imagen `mail_logo.png` de `www.google.com`.

****Total: 3 conexiones TCP**** son necesarias para cargar completamente la página usando HTTP/1.0: una para el HTML principal y una para cada una de las imágenes solicitadas automáticamente por la página.

Problema 9

Suponga la siguiente página escrita en HTML que reside en el servidor `www.fcen.uba.ar`:

```
<html>  
<head>  
<title>Facultad de Ciencias Exactas y Naturales</title>  
<link rel="stylesheet" type="text/css" href="style.css" />  
</head>  
<body>  
<div>  
  
<a href="avsearch.php">  </a>  
</div>  
<div>  
<form name="searchform" action="search">  
<label>Buscar</label>  
<input name="SearchableText" type="text" title="Buscar en el Sitio" />  
<input type="image" src="search_icon.gif" />  
</form>  
</div>  
</body>  
</html>
```

- a. ¿Cuánto tiempo en términos de RTTs (Round-Trip Time) transcurrirá como mínimo, hasta transferir la totalidad de la información en HTTP/1.0?

b. ¿Y en HTTP/1.1?

a. HTTP/1.0:

Cada recurso (HTML, CSS, imágenes) requiere una nueva conexión TCP, incluyendo el handshake de tres vías y la transferencia. Por lo tanto:

- **Total en HTTP/1.0**: 5 recursos = 5 conexiones = (5×2) RTTs = **10 RTTs** mínimo para transferir todo.

b. HTTP/1.1:

Con conexiones persistentes, se reduce el número de RTTs necesarios porque se pueden reutilizar las conexiones para múltiples recursos.

- **Total en HTTP/1.1**: 1 RTT para el handshake inicial + 1 RTT para la solicitud del HTML principal, y luego posiblemente todos los demás recursos en el mismo RTT si se usan bien las conexiones persistentes = **2 RTTs** en el mejor caso, aunque el número real puede variar dependiendo de cómo se manejen las solicitudes en paralelo.