

Redes de Computadoras

Santiago Botta

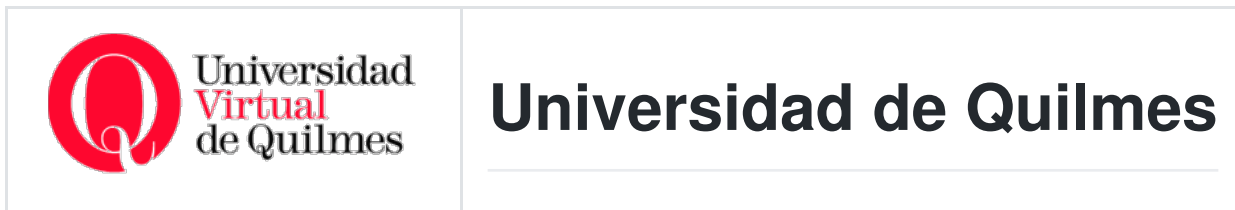
Práctica IV - HTTP/WWW

- Para el Problema 1 se utilizó éste [logo](#), dado que el ofrecido en el enunciado ya no está disponible.

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:

Maqueta 1



Tecnicatura en Programación Informática

Materias del núcleo básico obligatorio:

1. Introducción a la Programación
2. Organización de Computadoras
3. Matemática I
4. Programación con Objetos I
5. Bases de Datos
6. Estructuras de Datos
7. Programación con objetos II

[Volver a la página principal de la UNQ](#)

Código HTML

```
<head>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
</head>
<body>
  <table>
    <tr>
      <td>
        <i><u>Materias del n&uacute;cleo b&aacute;sico obliga


  <ol>
    <li>Introducci&ocute;n a la Programaci&ocute;n</li>
    <li>Organizaci&ocute;n de Computadoras</li>
    <li>Matem&aacute;tica I</li>
    <li>Programaci&ocute;n con Objetos I</li>
    <li>Bases de Datos</li>
    <li>Estructuras de Datos</li>
    <li>Programaci&ocute;n con objetos II</li>
  </ol>

  <table>
    <tr>
      <td style="width:50%"></td>
      <td><a href="http://www.unq.edu.ar/"><u>Volver a la p&aacute;gina principal

    </td>
  </tr>
</table>
```

```
</body>
```

Maqueta 2



Universidad
Virtual
de Quilmes

Universidad de Quilmes

Tecnicatura en Programación Informática

Materias del núcleo avanzado obligatorio:

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.

| Materia | Horas semanales | Carga horaria | Créditos |
|--------------------------|-----------------|---------------|----------|
| Redes de Computadoras | 6 | 108 | 12 |
| Sistemas Operativos | 6 | 108 | 12 |
| Programación Concurrente | 4 | 72 | 8 |
| Matemática II | 4 | 72 | 8 |

[Volver a la página principal de la UNQ](#)

Código HTML

```
<head>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
</head>
```

```

<body>
  <table>
    <tr>
      <td>
        <i><u>Materias del n&uacute;cleo avanzado obligatorio

  <p style="white-space: pre">&#9;&#9;Las materias del n&uacute;cleo avanzado obli
  estudiante. Para todas las materias incluidas en la siguiente tabla, el r&eacute;
  cuatrimestral, y la modalidad es presencial.</p>

  <table>
    <thead>
      <tr>
        <td><b>Materia</b></td>
        <td><b>Horas semanales</b></td>
        <td><b>Carga horaria</b></td>
        <td><b>Créditos</b></td>
      </tr>
    </thead>
    <tbody>
      <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>

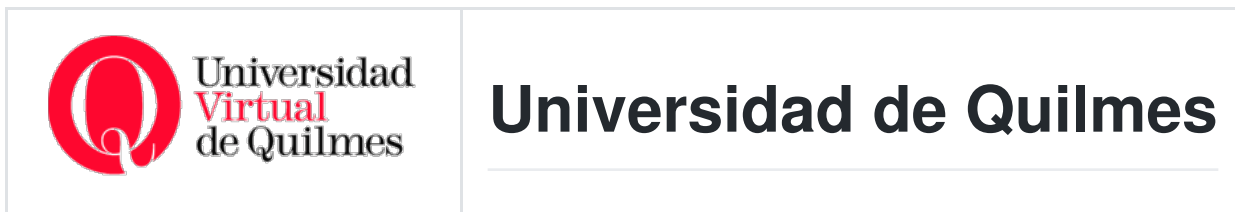
```

```

        <td>Programaci&oacute;n Concurrente</td>
        <td>4</td>
        <td>72</td>
        <td>8</td>
    </tr>
    <tr>
        <td>Matem&aacute;tica II</td>
        <td>4</td>
        <td>72</td>
        <td>8</td>
    </tr>
</tbody>
</table>
<table>
    <tr>
        <td style="width:50%"></td>
        <td><a href="http://www.unq.edu.ar/"><u>Volver a la p&aacute;gina principal
    </tr>
</table>
</body>

```

Maqueta 3



Tecnicatura en Programación Informática

Asignaturas del núcleo complementario:

Las materias del núcleo complementario permiten orientar al estudiante hacia un per determinado dentro del universo amplio y cambiante de los proyectos de desarrollo de software.

| | |
|--|---------------------------------------|
| | <input type="button" value="Buscar"/> |
|--|---------------------------------------|

[Volver a la página principal de la UNQ](#)

Código HTML

```
<head>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <script>
    function buscarMaterias() {
      http({
        method: 'POST',
        url: http://www.sistemas.unq.edu/cgi-bin/buscar-materias.exe
      })
      .then(function(res) {
        ...
      })
    }
  </script>
</head>
<body>
  <table>
    <tr>
      <td>
        <i><u>Asignaturas del n&uacute;cleo complementario:</

  <p style="white-space: pre">&#9;&#9;Las materias del n&uacute;cleo complementari
determinado dentro del universo amplio y cambiante de los proyectos de desarrollo
software.</p>

  <table>
```

```

        <tr>
            <td style="width:85%"><input style="width:100%" /></td>
            <td><button onclick="buscarMaterias)"></td>
            <td><a href="http://www.unq.edu.ar/"><u>Volver a la p&aacute;gina principal
        </tr>
    </table>
</body>

```

Problema 2

El protocolo HTTP permite hacer distintos tipos de pedidos para recursos de un determinado dominio. Escriba los Requests HTTP 1.1 que permitan obtener los siguientes pedidos al sitio web del departamento de computación:

1. **El recurso /**
2. **Encabezado del recurso /tdc**
3. **El recurso /logo.jpg si no fue modificado desde una determinada fecha**

1. Se solicita el recurso ubicado en / utilizando el protocolo HTTP 1.0 de la RFC 1945

GET / HTTP/1.0

2. Se solicita el recurso ubicado en /tdc utilizando el protocolo HTTP 1.0 de la RFC 1945

HEAD /tdc HTTP/1.0

3. Se solicita el recurso ubicado en /logo.jpg utilizando el protocolo HTTP 1.0 de la RFC 1945, con el encabezado (headers) correspondientes de manera que el recurso sea solicitado solo si no fue modificado desde el Lunes 20 de Abril del 2020 a las 4:20:00hs GMT

```
GET /logo.jpg
```

```
If-Unmodified-Since: "Mon", "20" "Apr", "2020", "04", "20", "00" GMT
```

Problema 3

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

```
telnet www.inta.gov.ar 80
GET / HTTP/1.1
Host: www.inta.gov.ar
```

```
telnet www.inta.gov.ar 80
GET / HTTP/1.1
Host: www.mercosurt.org.ar
```

En ambas secuencias de comando se está solicitando el recurso / de la misma fuente `www.inta.gov.ar`. Sin embargo, la diferencia es que se utilizan distintos valores para el encabezado `Host`, en la primera se realiza utilizando un encabezado `Host` con valor `www.inta.gov.ar`, en la segunda el valor del encabezado `Host` es `www.mercosurt.org.ar`.

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 diferentes deben estar siendo atendidos?
4. ¿Y si fueran 150.000.000 de clientes?

1. Con un servidor web es suficiente, las n conexiones simultáneas se harían hacia ese mismo servidor.
2. Una sola IP, para el servidor web
3. Los clientes serían atendidos en el puerto 80.

4. Si fueran 150.000.000 de clientes el servidor seguramente se estrese de pedidos. Sería ideal contar con más servidores. Sabiendo que un servidor soporta n cantidad de clientes habría que deducir cuántos servidores harían falta ($\text{cantidad-de-clientes} / \text{clientes-soportados-por-servidor}$). En ese caso se necesitaría asignar una IP distinta por servidor.

Problema 5

Interprete la salida del comando realizado desde una PC cualquiera conectada a Internet:

```
$ telnet www.inta.gov.ar 80
HEAD / HTTP/1.1
Host: www.inta.gov.ar

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
```

Se realiza un pedido http utilizando el programa `telnet` a la dirección `www.inta.gov.ar` en el puerto 80, solicitando el recurso `/` utilizando el protocolo HTTP 1.1 y asignando el valor `www.inta.gov.ar` al encabezado `Host`.

La salida del comando nos dice:

- En el `http status code` que el pedido fue correctamente devuelto (200, OK)
- El encabezado `Accept-Ranges` nos dice que el servidor soporta requests parciales, en particular en la unidad de `bytes`. Cuando existe el encabezado `Accept-Range` el navegador podría reanudar una descarga interrumpida en lugar de iniciarla desde el principio.
- El encabezado `Date` anuncia el momento (`Thu, 24 Mar 2011 23:15:43 GMT`) en el que el mensaje de respuesta fue originado.

- Content-Length indica la longitud (286) del mensaje de respuesta
- Content-Type especifica el tipo de formato (text/html) que utilizó el server para devolver el contenido de la respuesta.
- El encabezado Last-Modified declara la fecha (wed, 06 Oct 2010 12:46:39 GMT) en la que el servidor supone que el recurso solicitado fue modificado por última vez.
- ETag es un identificador para la versión (749338855465cb1:0") del recurso solicitado. Se utiliza para optimizar cache y ahorrar ancho de banda en la red.
- El encabezado Server describe el software (Microsoft-IIS/7.0) utilizado por el servidor de origen del recurso para devolver una respuesta.
- X-Powered-By es un encabezado no obligatorio que, en éste caso, denota la tecnología (ASP.NET) que respalda a la aplicación web o software del servidor de origen.

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 se realizan si se utiliza HTTP/1.0? ¿Y se utiliza HTTP/1.1?

Teniendo en cuenta que recién en HTTP/1.1 las conexiones pudieron comenzar a ser reutilizadas, se puede deducir que:

- En HTTP/1.0 se realizaría una conexión por cada uno de los recursos solicitados, es decir, 7 conexiones.
- EN cambio, utilizando el protocolo HTTP/1.1 las conexiones a un mismo servidor pueden ser reutilizadas para obtener recursos empujados dentro del documento original solicitado. De ésta manera podemos decir que se realizarían 3 conexiones, una para cada servidor.

Fuente: [Mozilla](#)

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 se realizarán usando HTTP/1.1 cuando se visite dicho sitio?

En HTTP/1.1 Se realizará una sola conexión TCP al servidor donde se encuentra index.html , luego se harán los pedidos HTTP correspondientes a los recursos .html y .jpg que sean necesarios.

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/">html hecho realmente fácil</h2>
    <a href="http://mail.google.com">
  </body>
</html>
```

¿Cuántas conexiones de nivel de transporte utiliza el navegador para transferir la totalidad de la información?

La cantidad de conexiones a nivel transporte dependerá estrictamente de la versión de protocolo que se utilice. Si el navegador utiliza HTTP/1.0 realizará una conexión para obtener el index.html de www.unq.edu.ar , luego otras dos conexiones para obtener los recursos /imagenes/logo_transp.gif y /imagenes/logos/mail_logo.png , respectivamente. Si el navegador utiliza HTTP/1.1 , realizará 1 sola conexión TCP , obtendrá el recurso index.html , luego los recursos solicitados en ese documento index.html y al finalizar cerrará la conexión TCP .

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>

```

1. ¿Cuánto tiempo en términos de RTTs transcurrirá como mínimo, hasta transferir la totalidad de la información en HTTP/1.0?

2. ¿Y en HTTP/1.1?

1. Asumo que **RTTs** se refiere a Round Trip Time. Asumo también que idealmente cada transferencia transcurre en 79ms y un *three-way TCP handshake* entre cliente-servidor demora 80ms en completarse. Por lo tanto, en una solicitud a `www.fcen.uba.ar`, utilizando HTTP/1.0, transcurrirán por lo menos $(80\text{ms} * 5\text{recursos}) + (5\text{recursos} * 79\text{ms}) = 795\text{ms}$, donde los recursos son:

- `index.html`
- `style.css`
- `searchline.png`
- `home.png`
- `search_icon.gif` Los RTTs que se analizan son los siguientes:
- cliente a resolver: 1RTT
- resolver a root con delegaciones a `ar.` y `uba.ar.`: 1RTT
 - se resuelve `index.html` en 2RTT

- los siguientes 4 recursos tardaran $1RTT$ cada uno gracias a la cache del resolver, entonces serían en total: $4RTT$. *Se determina entonces que en términos de RTTs, tomarán $6RTT$ en resolverse la web solicitada con todos sus objetos.*
2. En el contexto de solicitudes con protocolo `HTTP/1.1` el tiempo de respuestas solicitudes será el mismo. La diferencia es que la transferencia de recursos ocurrirá en una sola conexión, de manera que la duración total será de al menos $80ms + 5recursos * 79ms = 475ms$. *En cuánto a solicitudes en `HTTP/1.1` , tomarán $2RTT$ en resolverse la web solicitada con todos sus objetos.*

Problema 10

Un host sale a la web a través de un Proxy. El usuario navega solicitando páginas web hospedadas en el servidor web que tiene otros recursos (i.e.: imágenes) y además presenta propagandas hospedadas en el servidor de ads. Los rtts para las conexiones se muestran en la siguiente figura:

Figura en Practica 4 World Wide Web

Calcule los tiempos de los siguientes requests asumiendo que la cache del proxy empieza vacía y que se van cacheando los objetos sin expirar a lo largo de los pedidos, y que no hay cache local en el host:

1. El recurso `index.html` del servidor web conteniendo a su vez los recursos `1.jpg`, `2.jpg`, `3.jpg` y `4.jpg` hospedados en el servidor web, y los recursos `1.gif`, `2.gif` y `3.gif` hospedados en el servidor de ads.
2. El recurso `comprar.php` del servidor web conteniendo a su vez los recursos `3.jpg`, `4.jpg` y `5.jpg` hospedados en el servidor web, y los recursos `2.gif` y `3.gif` hospedados en el servidor de ads.
3. El recurso `gracias-por-comprar.html` del servidor web conteniendo a su vez los recursos `1.jpg`, `4.jpg` y `6.jpg` hospedados en el servidor web, y los recursos `1.gif` y `3.gif` hospedados en el servidor de ads.

Asumo que todas las conexiones se realizan con protocolo `HTTP/1.1` y que el proxy no tiene que resolver las direcciones del servidor Web y Servidor de ads con otros servidores DNS.

Para calcular los tiempos que demoran las requests, asumo que los archivos `.html` , `.php` , `.jpg` y `.gif` demoran 50ms.

1. Se realizan los siguientes pedidos:

- solicitud de conexión, usuario a proxy: 10ms
- solicitud de conexión, proxy a servidor web: 300ms
 - se recuperan los archivos index.html , 1.jpg , 2.jpg , 3.jpg y 4.jpg del servidor web: 250ms
- solicitud de conexión, usuario a proxy: 10ms
- solicitud de conexión, proxy a servidor de ads: 400ms
 - se recuperan los archivos 1.gif , 2.gif y 3.gif del servidor de ads: 150ms
- **Total:** 10ms + 300ms + 250ms + 10ms + 400ms + 150ms = 1120ms

2. Se realizan los siguientes pedidos:

- solicitud de conexión, usuario a proxy: 10ms
- solicitud de conexión, proxy a servidor web: 300ms
 - se recuperan los archivos comprar.php y 5.jpg del servidor web: 100ms
 - se recuperan los archivos 3.jpg y 4.jpg de la caché de Proxy: 0ms
- solicitud de conexión, usuario a proxy: 10ms
- solicitud de conexión, proxy a servidor de ads: 400ms
 - se recuperan los archivos 2.gif y 3.gif de la caché de Proxy: 0ms
- **Total:** 10ms + 300ms + 100ms + 0ms + 10ms + 400ms + 0ms = 820ms

3. Se realizan los siguientes pedidos:

- solicitud de conexión, usuario a proxy: 10ms
- solicitud de conexión, proxy a servidor web: 300ms
 - se recuperan los archivos gracias-por-comprar.html y 6.jpg del servidor web: 100ms
 - se recuperan los archivos 1.jpg y 4.jpg de la caché de Proxy: 0ms
- solicitud de conexión, usuario a proxy: 10ms
- no se realiza conexión a servidor de ads, se reutilizan los objetos 1.gif y 3gif en caché de Proxy.
- **Total:** 10ms + 300ms + 100ms + 0ms + 10ms = 420ms

Fuente y material de investigación utilizado para ésta práctica además del libro Redes de Computadoras de Tanenbaum:

- **Redes de Computadoras - A. S. Tanenbaum**

- [Making HTTP requests via telnet](#)
- [Evolución de HTTP - Mozilla](#)
- [Evolución de HTTP - Medium](#)
- [Http headers - Mozilla Developers](#)
- [TCP 3-way handshake](#)
- [Questions about Round Trip Time](#)