



Facultad de Estudios Superiores

Acatlán

WEB

José Antonio Coria Fernández
Email: diplomadobd@gmail.com

Contenido

- Interacción Cliente-Servidor
- Aplicaciones en Internet
 - DNS
 - WWW
- PHP y JSP

Como hacen contacto dos aplicaciones

- Una aplicación (programa pasivo)
 - Comienza ejecutándose primero
 - Espera pasivamente en una ubicación acordada de antemano
- Otra aplicación (programa activo)
 - Inicia su ejecución posteriormente
 - Activamente contacta al primer programa
- A lo anterior se conoce como interacción cliente-servidor

- Utilizado por todas las aplicaciones en red
- Programa pasivo llamado servidor
- Programa activo llamado cliente

Características de un cliente

- Programa de aplicación arbitrario
- Se convierte en cliente temporalmente
- Puede desempeñar otros cálculos
- Invocado directamente por el usuario
- Se ejecuta localmente en la computadora del usuario
- Activamente inicia contacto con un servidor
- Contacta un servidor a la vez

Características de un Servidor

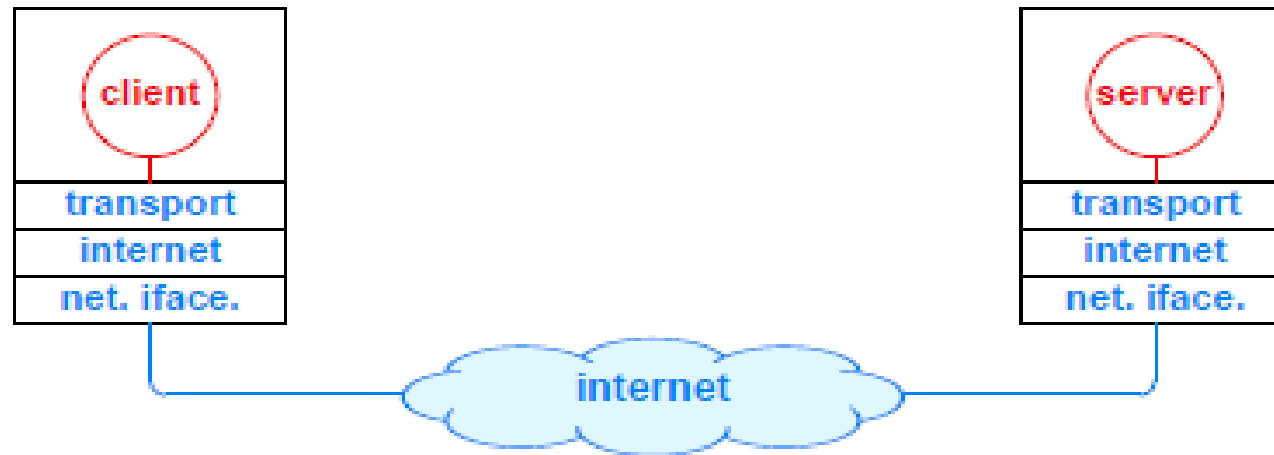
- Programa privilegiado de propósito especial
- Dedicado a proporcionar un servicio
- Puede manejar múltiples clientes remotos simultáneamente
- Invocado directamente cuando el sistema arranca
- Se ejecuta por siempre
- Require de una computadora y sistema operativo robusto
- Espera pasivamente por el contacto de un cliente
- Acepta solicitudes de clientes arbitrarios

- El flujo de los datos puede ser:
 - Del cliente al servidor únicamente
 - Del servidor al cliente solamente
- El protocolo de aplicación determina el flujo
- Escenarios típicos
 - Cliente envía solicitud
 - Servidor envía respuesta

Aunque el contacto lo inicia el cliente, el flujo de la información en cualquier dirección o en ambas entre un cliente y un servidor. Muchos servicios organizados por el cliente envían una o más solicitudes y el servidor regresa una respuesta por cada solicitud.

Clientes, Servidores

- Clientes y servidores son programas de aplicación.

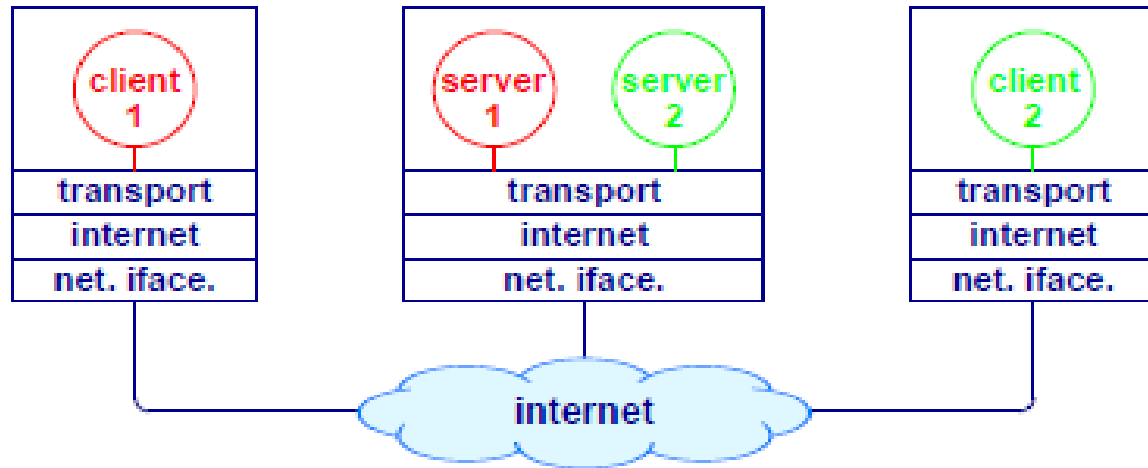


- Hechos
 - El servidor opera como otra aplicación
 - Emplea el CPU para ejecutar instrucciones
 - Desempeña operaciones de E/S
 - El esperar que los datos lleguen por la red no requiere tiempo de CPU
- Consecuencias
 - El programa servidor solo utiliza el CPU cuando da respuesta a una solicitud

Múltiples servicios

- Se pueden manejar múltiples servidores en una sola computadora
- Los servidores solo utilizan el procesador cuando manejan solicitudes
- Hardware robusto es requerido para manejar varios servicios simultáneamente.

Múltiples Servidores



- Cada servidor ofrece un servicio
- Un servidor puede manejar múltiples clientes

Identificando un Servicio

- Un número de puerto de protocolo es utilizado
- Cada servicio prestado cuenta con un número de puerto único, P
- Servidor
 - Informa al SO que esta utilizando un puerto P
 - Espera por el arribo de una solicitud
- Cliente
 - Formula una solicitud
 - Envía la solicitud al puerto P en una computadora servidor

La cuestión de los puertos

- * Los protocolos de transporte asignan a cada servicio un identificador de puerto único.
- * Un servidor debe especificar el identificador cuando inicia su ejecución.
- * Un cliente debe especificar el identificador cuando solicita al protocolo de transporte contactar al servidor.
- * El protocolo de transporte en la computadora servidor utiliza el identificador para direccionar una solicitud de entrada al servidor correcto.

En la práctica los puertos

- El número de puerto del protocolo es utilizado para identificar servicios
- Necesitan de una numeración uniforme
 - Para permitir que clientes arbitrarios contacten servidores en máquinas arbitrarias
 - Para evitar mecanismos inventados de asistencia de directorios
- Número de puerto
 - Fijados por las entidades de estándares (IANA)
 - Uniformes por toda la Internet

- El número de puerto esta dividido en tres categorías:
 - Los bien conocidos , 0-1023
 - Los puertos registrados 1024-49151
 - Los puertos dinámicos y / o privados 49152-65535

- Servidor secuencial
 - También llamado iterativo
 - Maneja una solicitud a la vez
- Servidor Concurrente
 - Puede manejar múltiples solicitudes a la vez
 - No hay espera

- Servidor concurrente
 - El servidor crea un nuevo hilo de control para manejar cada solicitud
 - El cliente solo espera por su solicitud para ser procesada
- Servidor secuencial
 - El cliente espera por todas las solicitudes previas que tienen que ser procesadas, así como por la respuesta a su solicitud.
 - Inaceptable para los usuarios si una solicitud larga bloquea solicitudes cortas

- La ejecución concurrente es fundamental en los servidores debido a que la concurrencia permite a múltiples clientes obtener un servicio dado sin tener que esperar a que el servidor termine la solicitud previa.
- En un servidor concurrente, el proceso servidor principal crea un nuevo proceso de servicio(hilo) para manejar cada cliente.

Protocolo de puertos y servidores concurrentes

- Problema aparente
- Un puerto asignado a cada servicio
- Un servidor concurrente tiene múltiples copias ejecutandose (hilos)
- Cliente y servidor deben interactuar
- El mensaje enviado al puerto del servidor, debe ser entregado a la copia correcta

Protocolo de puertos y servidores concurrentes

- Solución al Problema: utilizar información acerca del cliente así como del servidor para la entrega de paquetes de entrada
- TCP utiliza cuatro puntos para identificar la conexión:
 - Dirección IP del servidor
 - Número de puerto del servidor
 - Dirección IP del cliente
 - Número de puerto del cliente

Aplicaciones de Internet

DNS

- Las comunicaciones en internet requieren de direcciones IP
- Los humanos preferimos utilizar nombres para las computadoras
- Se cuenta con sistemas automatizados para hacer la traducción de nombres a direcciones
- Se conocen como Sistemas de Nombre de Dominios (sus siglas en inglés ,DNS)

Aplicaciones de Internet

Funcionalidad DNS

- Proporciona
 - Nombre de computadora
- Regresa
 - Dirección de Internet de la computadora (Dirección IP)
- Método
- Búsqueda distribuida
 - El cliente contacta un servidor o servidores como sea necesario

Aplicaciones de Internet

Sintaxis DNS

- Segmentos alfanúmericos separados por puntos
- Ejemplos:
 - www.acatlan.unam.mx
 - www.google.com
 - www.hotmail.com
- La parte más significativa a la derecha
- El primer nivel es país: mx
- El segundo nivel es la organización: unam
- El tercer nivel es una división dentro de la organización: acatlan
- El cuarto nivel puede ser: una computadora individual o bien una subdivisión de la organización.: www

Aplicaciones de Internet

Obtención DNS

- Organización
 - Elegir un nombre deseado
 - Debe ser único
 - Registrarlo con una autoridad central
 - Ubicarlo bajo uno de los dominios de nivel más alto
- Nombres sujetos a leyes internacionales debido a
 - Derechos de autor (copyright)
 - Marcas registradas (trademark)

Aplicaciones de Internet

Dominios de nivel más alto

<i>Domain Name</i>	<i>Assigned To</i>
<i>com</i>	<i>Commercial organization</i>
<i>edu</i>	<i>Educational institution</i>
<i>gov</i>	<i>Government organization</i>
<i>mil</i>	<i>Military group</i>
<i>net</i>	<i>Major network support center</i>
<i>org</i>	<i>Organization other than those above</i>
<i>arpa</i>	<i>Temporary ARPA domain (still used)</i>
<i>int</i>	<i>International organization</i>
<i>country code</i>	<i>A country</i>

Aplicaciones de Internet WEB

Tres componentes básicos concebidos por Tim Berners-Lee comprendiendo la esencia de la tecnología Web:

- Un lenguaje de marcaje para dar formato a documentos de hipertexto
- Un esquema de notación uniforme para direccionar recursos accesibles en la red.
- Un protocolo para transportar los mensajes sobre la red.

Aplicaciones de Internet

Componentes Web

- Navegador
- Servidor Web
- Enlaces de hipermedia
- Representación de documentos
- Protocolo de transferencia

Aplicaciones de Internet

Componentes Web

- Navegador
 - Programa de aplicación
 - Interfaz para el usuario en la Web
 - Se convierte en el cliente Web que trae información del servidor Web
 - Despliega información al usuario

Aplicaciones de Internet

Componentes Web

- Servidor Web
 - Programa en ejecución
 - Almacena un conjunto de documentos Web
 - Responde a las solicitudes del navegador al enviar una copia del documento

Aplicaciones de Internet

Componentes Web

- Hipermedia
 - Documento web que contiene una mezcla de:
 - Texto
 - Imágenes
 - Áreas en pantalla o asociadas a objetos que son seleccionables y direccionan a otras páginas Web (ligas o enlaces)
 - Ventaja: Puede direccionar a documentos en otra computadora
 - Desventaja: Se puede convertir en inválido

Aplicaciones de Internet

Componentes Web

- Documento Web, llamado Página Web
- Una página web por archivo
- Puede contener : imágenes, texto
- Texto estándar:
 - Especifica contenido y diseño
 - Representación ASCII legible
 - Conocido como Lenguaje de Marcas de hipertexto (HTML)
 - Proporciona pautas generales de diseño
 - No especifica una ubicación exacta o formato

Aplicaciones de Internet Web

- Los documentos web emplean la representación de lenguaje de marcaje de hipertexto. En lugar de especificar un formato detallado de documento, el HTML permite que éste tenga guías generales de presentación y que el visualizador (navegador) se encargue de los detalles. En consecuencia, dos navegadores pueden representar de manera diferente un documento HTML

Aplicaciones de Internet Web

- Cada documentos HTML se divide en dos partes principales: una cabecera seguida de un cuerpo.
- Etiquetas dan guías de desplegado
- Las etiquetas aparecen en pares
- Formato de etiqueta
 - Etiqueta de inicio
`<REFERENCIA>`
 - Etiqueta de finalización
`</REFERENCIA>`

Ejemplo:

<HTML>

<HEAD>

<TITLE>

Titulo del documento

</TITLE>

</HEAD>

<BODY>

Cuerpo del documento

</BODY>

</HTML>

Aplicaciones de Internet Web

- El código HTML es libre de formato
- El ejemplo previo es equivalente a:

```
<HTML><HEAD><TITLE>titulo del documento  
</TITLE></HEAD><BODY>cuerpo del documento  
</BODY></HTML>
```

Etiquetas HTML

- Comienzo de párrafo

`<P>`

- Romper línea (forzar una nueva línea)

`
`

- Encabezado principal (fuente marcada y Grande)

`<H1> ...TEXTO... </H1>`

- Siguiendo encabezado (siguiendo tamaño)

`<H2> ...TEXTO.. </H2>`

Enlaces o ligas a otras páginas

- Representación simbólica
- Embebido en el documento HTML
- El navegador (visualizador):
 - Esconde el texto del enlace al usuario
 - Asocia enlaces con objetos en la página
 - Convierte objetos seleccionables

* Llamado Localizador de Recursos Uniforme

Aplicaciones de Internet Web

Formato General de URL (esquema de notación uniforme para direccionar a recursos accesibles sobre la red)

esquema : // computadora : puerto / nombre_archivo

- Esquema: protocolo http o ftp
- computadora: www.acatlan.unam.mx o bien 132.248.80.197
- Puerto: es un valor opcional, de forma predeterminada se hace referencia al 80
- nombre_archivo: index.html (o bien /ruta/archivo)

Fundamentos del Protocolo HTTP

- El protocolo HTTP emplea el paradigma solicitud/respuesta, esto significa que un programa cliente HTTP envía una solicitud de mensaje al servidor HTTP, el cual regresa un mensaje de respuesta HTTP.
- La estructura de un mensaje de solicitud y respuesta es similar a la de un mensaje de correo electrónico, que consiste en un grupo de líneas que contienen encabezado de mensaje, seguido de líneas en blanco y finalmente un cuerpo del mensaje
- HTTP es un protocolo sin estado. El tiempo de vida de una conexión es un simple intercambio de solicitud y respuesta. Esto significa que no hay manera de mantener la persistencia de la información acerca de una “sesión” de interacciones sucesivas entre el cliente y el servidor.

Aplicaciones de Internet Web

Fundamentos del Protocolo HTTP

- Por el contrario, cuando un protocolo soporta “estado”, significa que proporciona para la interacción del cliente y el servidor una secuencia de comandos denominados sesión, hasta que la conexión es terminada.

Aplicaciones de Internet Web

Estructura de un mensaje de solicitud HTTP

- Como ejemplo tomamos la solicitud a una página estática que reside en un Servidor Web. El usuario ingresa el URL <http://www.ejemplo.com/sj/index.html> en el navegador (o da clic a una liga). Esta acción es traducida por el navegador en una solicitud HTTP con la siguiente estructura:

METHOD /sj/index.html (ruta_al_recurso) HTTP/1.1 (num_version)

Host: www.ejemplo.com

Estructura de un mensaje de respuesta HTTP

HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 9934

...

<HTML>

<HEAD>

<TITLE>Pagina Ejemplo</TITLE>

</HEAD>

<BODY BGCOLOR="#ffffff">

<H2 ALIGN="center">Bienvenidos a la WEB</H2>

...

</H2>

</BODY>

</HTML>

Aplicaciones de Internet Web

Métodos de solicitud

- Imponen restricciones en la estructura del mensaje, y especificaciones de como el servidor debe procesar la solicitud.
- Los más comunes en uso GET, POST, los menos comunes HEAD, PUT,DELETE,TRACE,OPTIONS y CONNECT.
- Método GET, una solicitud de este tipo no tiene cuerpo. Ejemplo:
Al llenar un formulario HTML para solilcitar una cotización en la bolsa

Método GET

<HTML>

<HEAD><TITLE>Formulario simple</TITLE></HEAD>

<BODY>

<H2>Formulario Simple</H2>

<FORM ACTION="http://finance.yahoo.com/q" METHOD="get">

<!-- <FORM ACTION="http://finance.yahoo.com/q" METHOD="POST"> -->

Industria: <INPUT TYPE="text" SIZE="25" NAME="cantidad" VALUE="">

<INPUT TYPE="submit" VALUE="CotizacionBolsa">

</FORM>

<BODY>

</HTML>

Método GET

- Si se ingresara la palabra 'YHOO' en el formulario anterior, el navegador construiría un URL compuesto del campo ACTION del formulario, seguido por una cadena de consulta que contiene todos los parámetros de entrada del formulario y los valores que les fueron asignados. El separador del URL de la cadena de consulta es un signo de interrogación (?). El parámetro es cantidad y el valor asignado YHOO
- El navegador muestra
<http://www.finance.yahoo.com/q?cantidad=YHOO>
- Y la solicitud enviada se ve así:

Método GET

GET /q?cuenta=YHOO HTTP/1.1

Host: finance.yahoo.com

User-Agent: Mozilla/4.75 [en] (WinNT; U)

Y la respuesta:

HTTP/1.0 200 OK

Date: Sat, 03 Feb 2001 22:48:35 GMT

Connection: close

Content-Type: text/html

Set-Cookie: B=9ql5kgct7p2m3&b=2;expires=Thu,15 Apr 2010 20:00:00 GMT;
path=/; domain=.yahoo.com

<HTML>

<HEAD><TITLE>Yahoo! Finance - YHOO</TITLE></HEAD>

Método POST

La principal diferencia es que la solicitud POST tiene un cuerpo, Y en este (el cuerpo del mensaje) envía los parámetros del formulario, en vez de colocarlos en el URL. El navegador solo muestra:

`http://www.finance.yahoo.com/`

`POST /q HTTP/1.1`

`Host: finance.yahoo.com`

`User-Agent: Mozilla/4.75 [en] (WinNT; U)`

`Content-Type: application/x-www-form-urlencoded`

`Content-Length: 6`

`cuenta=YHOO`

— El mensaje de respuesta es exactamente al mismo del GET

PHP

Sus siglas son un acrónimo recursivo a Hypertext Preprocessor. Permite a los desarrolladores , código embebido dentro de plantillas HTML, con un lenguaje similar a PERL y shell de UNIX. El objeto fuente es estructurado en una página HTML, pero dinámicamente la generación de contenido es programado.

Ejemplo:

```
<html>
<body>
    <?php
        for(i=0;i<=4; i++){ echo "hola mundo";}
    ?>
</body>
</html>
```

PHP (1/2)

<HTML>

<HEAD><TITLE>Ejemplo de conexion Oracle simple</TITLE></HEAD>

<BODY BGCOLOR="#FFFFFF">

<CENTER>

Empleados

<?php

```
if( $conn = oci_connect("scott", "tiger", "127.0.0.1/orcl") ){
```

```
$cad = 'select empno,ename from scott.emp';
```

```
$cons = oci_parse($conn, $cad);
```

```
oci_execute($cons);
```

```
echo "<TABLE CELSPACING=\"0\" CELLPADDING=\"3\" BORDER=\"1\">";
```

```
echo "<TR><TH>Identificador</TH><TH>Nombre</TH></TR>";
```


PHP (2/2)

```
while ($res = oci_fetch_array($cons, OCI_ASSOC)) {  
    echo "<tr>";  
    echo "<td>".$res['EMPNO']."</td>";  
    echo "<td>".$res['ENAME']."</td>";  
    echo "</tr>";  
}  
echo "</table>";  
}else{  
    echo "no se pudo conectar";  
}  
?>  
</CENTER>  
</BODY>  
</HTML>
```


JSP

El Servidor de Páginas Java (Java Server Pages), al igual que PHP el soporte de JSP fue implementado a través de un pre-procesador que devuelve objetos de la página con bloques de código embebido dentro de un código fuente servlet.

Ejemplo:

```
//<%@ import= java.io.*" %> ejemplo para importar clases
```

```
<html>
```

```
<body>
```

```
<% for (i=0;i<=4;i++){ out.println("hola mundo");}
```

```
%>
```

```
</body>
```

```
</html>
```

JSP

El Servidor de Páginas Java (Java Server Pages), al igual que PHP el soporte de JSP fue implementado a través de un pre-procesador que devuelve objetos de la página con bloques de código embebido dentro de un código fuente servlet.

Ejemplo:

```
//<%@ import= java.io.*" %> ejemplo para importar clases
```

```
<html>
```

```
<body>
```

```
<% for (i=0;i<=4;i++){ out.println("hola mundo");}
```

```
%>
```

```
</body>
```

```
</html>
```

JSP

Ejemplo código Servlet: (1/2)

```
package jsp. myapp ;  
import java.io.* ;  
import java.util.* ;  
import javax.servlet.* ;  
import javax.servlet.http.* ;  
import javax.servlet.jsp.* ;  
public class mypage extends HttpJspBase {  
    private CustomObject myObject;  
    public void jspService(HttpServletRequest req, HttpServletResponse  
    resp)  
    {  
        ServletConfig config = getServletConfig() ;  
        ServletContext application = config.getServletContext() ;
```

JSP

Ejemplo código Servlet: (2/2)

Object page = this ;

PageContext pageContext =

JspFactory.getDefaultFactory().getPageContext(this, req, resp,
null, true, 8192, true) ;

JspWriter out = pageContext.getOut() ;

HttpSession session = request.getSession(true) ;

```
for(int i = 0; i <= 4; i++) {  
    out.print("Hola Mundo") ;  
}  
  
}  
  
}
```

JSP (1/4)

```
<%@ page import="java.sql.*" %>
```

```
<HTML>
```

```
<HEAD><TITLE>Ejemplo de conexion Oracle simple</TITLE></HEAD>
```

```
<BODY BGCOLOR="#FFFFFF">
```

```
<CENTER>
```

```
<B>Empleados</B>
```

```
<BR><BR>
```

```
<%
```

```
Connection conn = null;
```

```
try
```

```
{
```

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
conn =
```

```
    DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:orcl","scott","tiger");
```


JSP (2/4)

```
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM scott.emp");
out.println("<TABLE CELLSPACING=\"0\" CELLPADDING=\"3\" BORDER=\"1\">");
out.println("<TR><TH>Identificador</TH><TH>Nombre</TH></TR>");
while(rs.next())
{
out.println("<TR>");
out.println("<TD>" + rs.getString("EMPNO") + "</TD>");
out.println("<TD>" + rs.getString("ENAME") + "</TD>");
out.println("</TR>");
}
out.println("</TABLE>");
}
catch(SQLException e){
```


JSP (3/4)

```
out.println("SQLException: (error de Conexion) " + e.getMessage() + "<BR>");
while((e = e.getNextException()) != null)
out.println(e.getMessage() + "<BR>");
}
catch(ClassNotFoundException e) {
out.println("ClassNotFoundException: (Clases importadas no las encuentra) " +
    e.getMessage() + "<BR>");
}
finally {
if(conn != null){
try{
conn.close();
}
}
```

Aplicaciones de Internet Web

JSP (4/4)

```
catch (Exception ignored) {}
```

```
}
```

```
}
```

```
%>
```

```
</CENTER>
```

```
</BODY>
```

```
</HTML>
```