

Sistemas distribuidos	introducción
Contenido presentación introducción tecnología web middleware servicios web conclusiones	bibliografía <ul style="list-style-type: none"> Sistemas Distribuidos. Conceptos y Diseño G. Coulouris et al. Addison Wesley, 2001 Temas 3, 4, 5 y 17 Service-Oriented Architecture: Concepts, technology and Design T. Erl Prentice Hall, 2005 Temas 3, 4 y 5 Computación Distribuida. Fundamentos y Aplicaciones M.L. Liu Addison Wesley, 2004 Libro completo

Sistemas distribuidos	introducción
Contenido presentación introducción tecnología web middleware servicios web conclusiones	contenidos <ul style="list-style-type: none"> Revisión de tecnologías Web <ul style="list-style-type: none"> Cliente: DHTML, ActiveX, applet, plug-ins Servidor: CGI, servlets, páginas activas, extensiones, servidor de aplicaciones Middleware <ul style="list-style-type: none"> Arquitectura y modelo de servicios Modelo de programación Idoneidad del modelo para desarrollar aplicaciones distribuidas Servicios Web <ul style="list-style-type: none"> Componentes básicos: SOAP, WSDL, UDDI SOA y Servicios Web

Sistemas distribuidos

tecnología web

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

tecnología web

Sistemas distribuidos

tecnología web

modelos básicos

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

Modelos básicos

Modelo C/S sobre Internet

```

graph LR
    Cliente[Cliente] -- "dir IP o DNS" --> Internet[Internet]
    Internet -- "dir IP o DNS" --> Servidor[Servidor]
    Servidor -- "Recurso" --> Cliente
    Servidor -- "servicio" --> Cliente
    Servidor -- "Know-How" --> Servidor
    
```

Sistemas distribuidos

tecnología web

modelos básicos

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

Modelos básicos

- Modelo Web Básico

Modelo C/S sobre Internet

Sistemas distribuidos

tecnología web

modelos básicos

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

Modelos básicos

- Modelo Web Básico

servicio HTTP básico

Método	Recurso	Versión	Cabecera
GET	/index.html	HTTP/1.0	

Sistemas distribuidos

tecnología web

modelos básicos

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

Modelos básicos

- | | |
|------|----------------------------------|
| GET | Solicitar contenido |
| HEAD | Solicitar únicamente la cabecera |
| POST | Envía datos al servidor |
| PUT | Envía contenido para almacenar |

Principales métodos HTTP

Línea

Cabecera

Cuerpo

POST /cgi/miAplicacion.cgi HTTP/1.0

Accept: */*

Connection: Keep-Alive

User-Agent: Generic

[línea en blanco]

Nombre=Paco&eMail=pmacia@dtic.ua.es

Ejemplo de solicitud HTTP

Sistemas distribuidos

tecnología web

modelos básicos

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

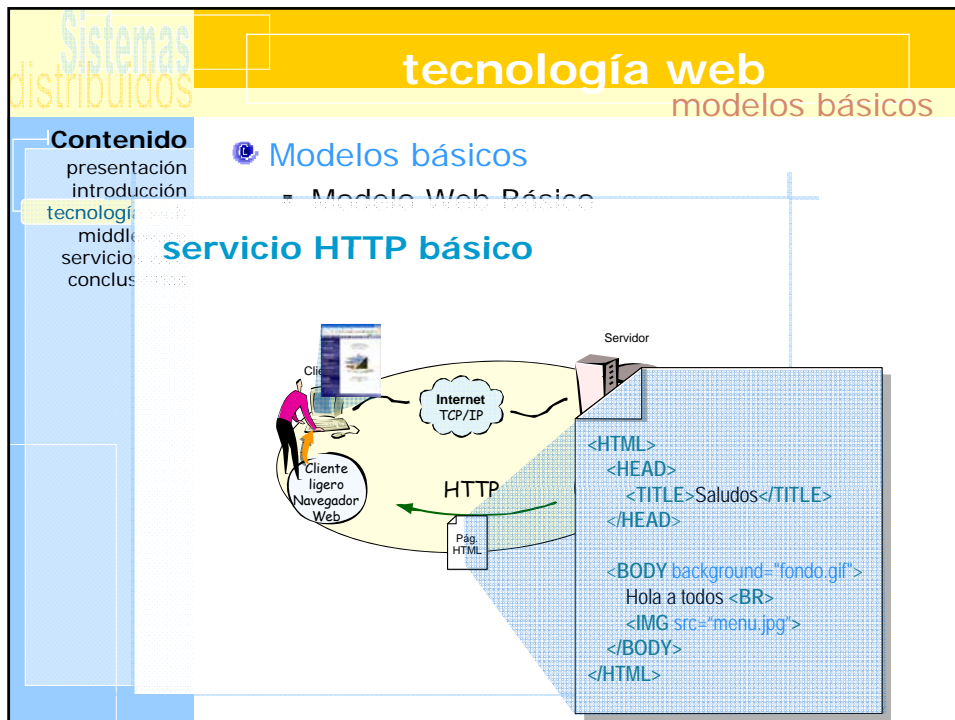
Modelos básicos

- Modelo Web Básico

servicio HTTP básico

```

graph LR
    Client[Cliente ligero Navegador Web] -- HTTP --> Internet((Internet TCP/IP))
    Internet --> Server[Servidor]
    Server -- "Pág. HTML" --> Client
    
```



Sistemas distribuidos

tecnología web

modelos básicos

Contenido
 presentación
 introducción
 tecnología web
 middle
 servicios web
 conclusiones

Modelos básicos
 ■ Modelo Web Básico

HTML (HiperText Markup Language)

- Documento de texto
- Etiquetas de formato
 - `<etiqueta [arg1 arg2 ...]> ... [</etiqueta>]`
- Referencias cruzadas
 - ` texto descriptivo `

MIME (Multipurpose Internet Mail Extension)

Sistemas distribuidos básicos

```
C:\WINDOWS\system32\cmd.exe
C:\> telnet www.dtic.ua.es 80
Trying www.dtic.ua.es...
Connected to www.dtic.ua.es
Escape character is '^]'.
GET /index.html HTTP/1.0

+HTTP/1.1 200 OK
Date: Thu, 04 Nov 2004 17:59:15 GMT
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Wed, 03 Nov 2004 13:01:02 GMT
ETag: "378029-902-3bd83f80"
Accept-Ranges: bytes
Content-Length: 256
Connection: close
Content-Type: text/html; charset=ISO-8859-1

<html>
<head>
<title>Tecnología Informática y Computación</title>
</head>
<body background="fondo.gif">
Hola a Todos<br>

</body>
</html>

Se ha perdido la conexión con el host.
C:\> _
```

Sistemas distribuidos **tecnología web** modelos básicos

Contenido
presentación
introducción
tecnología web
middleware
servicios web
conclusiones

Modelos básicos

- Modelo Web Básico

Línea	POST /cgi/miAplicacion.cgi HTTP/1.0
Cabecera	Accept: /*/* Connection: Keep-Alive User-Agent: Generic [línea en blanco]
Cuerpo	Nombre=Paco&eMail=pmacia@dtic.ua.es

Ejemplo de solicitud HTTP

Sistemas distribuidos

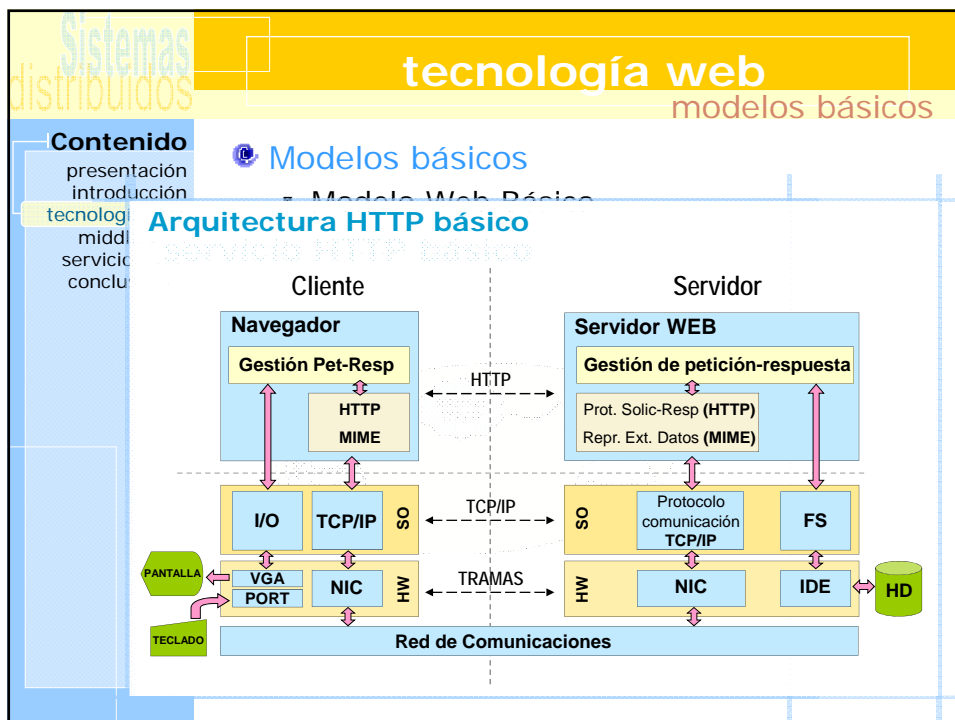
tecnología web

modelos básicos

Contenido
 presentación
 introducción
tecnología web
 middleware
 servicios web
 conclusiones

1xx Mensajes de información 100 Continua 101 Cambio de protocolo 2xx Operación exitosa 200 Ok 201 Creado 202 Aceptado 203 Información no oficial 204 Sin Contenido 205 Contenido para reset 206 Contenido parcial 3xx Redirección hacia otro URL 300 Múltiples posibilidades 301 Mudado permanentemente 302 Encontrado 303 Véa otros 304 No modificado 305 Utilice un proxy 307 REDirección temporal	4xx Error por parte del cliente 400 Solicitud incorrecta 401 No autorizado 402 Pago requerido 403 Prohibido 404 No encontrado 405 Método no permitido 406 No aceptable 407 Proxy requerido 408 Tiempo de espera agotado 409 Conflicto 410 No mpas disponible 411 Requiere logitud 412 Falló precondition 413 Entidad de solicitud demasiado larga 414 URI de solicitud demasiado largo 415 Tipo de medio no soportado 416 Rango solicitado no disponible 417 Falló expectativa 5xx Error por parte del servidor 500 Error interno 501 No implementado 502 Pasarela incorrecta 503 Servicio no disponible 504 Tiempo de espera de la pasarela agotado 505 Versión de HTTP no soportada
---	--

Códigos de respuesta HTTP



Sistemas distribuidos

tecnología web

modelos básicos

Contenido
 presentación
 introducción
tecnología web
 middleware
 servicios web
 conclusiones

Modelos básicos

- Modelo Web Básico
- Modelo CGI (Common Gateway Interface)

Método	Ruta-Aplicación-Parámetros	Versión	Cabecera
GET	/cgi-bin/buscar.cgi?nombre=Paco%20Maciá	HTTP/1.0	

Variable de Entorno	Valor
REQUEST_METHOD	GET
QUERY_STRING	/cgi-bin/buscar.cgi?nombre=Paco%20Maciá
CONTENT_LENGTH	39
SCRIPT_NAME	/cgi-bin/buscar.cgi
SERVER_PORT	80

Sistemas distribuidos

tecnología web

modelos básicos

Contenido
 presentación
 introducción
tecnología web
 middleware
 servicios web
 conclusiones

```

/* Programa C utilizado como programa CGI  *
 * generando como salida una página html  *
 * con el mensaje "Hola a todos"          */

#include <stdio.h>

main(int argc, char *argv[]) {

    printf("Content-type: text/html\n\n");
    printf("<HTML>\n");
    printf("<HEAD>\n");
    printf("<TITLE>HOLA MUNDO</TITLE>\n");
    printf("</HEAD>\n");
    printf("<BODY>\n");
    printf("<H1>Hola a Todos!</H1>\n");
    printf("</BODY>\n");
    printf("</HTML>\n");

} /* de main */

```

Código fuente del CGI 'Hola mundo' en C

Sistemas distribuidos

tecnología web

modelos básicos

Contenido
 presentación
 introducción
tecnología web
 middleware
 servicios web
 conclusiones

Modelos básicos

```
#!/usr/local/sh
# Comenzamos indicando quién debe interpretar el script.

# Sencillo script de shell que genera por la salida
# estándar una página HTML con el mensaje deseado.

# Generamos la cabecera de la respuesta
print "Content-type: text/html"
print

# Generamos el cuerpo de la respuesta
print "<HTML>"
print "<HEAD>"
print "<TITLE>HOLA MUNDO</TITLE>"
print "</HEAD>"
print "<BODY>"
print "<H1>¡Hola a Todos!</H1>"
print "</BODY>"
print "</HTML>"
```

Guion de comandos (script de shell) compatible con la mayoría de shells UNIX (ksh, bsh, bash) que actúa como un programa CGI

Sistemas distribuidos

tecnología web

Contenido
 presentación
 introducción
tecnología web
 middleware
 servicios web
 conclusiones

```
Int main (void)
{
    char metodo[100];
    char argumentos[1000];
    int tam;

    /* Obtengo el metodo */
    strcpy(metodo, getenv("REQUEST_METHOD"));
    if (strcmp(metodo, "GET") == 0)
    {
        /* Los argumentos están en la variable de entorno QUERY_STRING */
        strcpy(argumentos, getenv("QUERY_STRING"));
    }
    else
    {
        if (strcmp(metodo, "POST") == 0)
        {
            /* Los argumentos están en la entrada estándar */
            tam = atoi(getenv("CONTENT_LENGTH"));
            fgets(argumentos, tam + 1, stdin);
        }
        else
            printf("Error. Metodo no soportado");
    }

    /* Cabecera */
    printf("Content-type: text/html\n");
    printf("\n");

    /* Cuerpo */
    printf("<HTML>\n");
    printf("<HEAD>\n");
    printf("<TITLE>Argumentos de entrada</TITLE>\n");
    printf("<BODY>\n");
    printf("Argumentos del CGI: %s\n", argumentos);
    printf("</BODY>\n");
    printf("</HTML>\n");
}
```

Ejemplo en C del tratamiento de las peticiones HTTP GET y POST en un CGI

tecnología web	
Nombre	Descripción
REQUEST_METHOD	Comando HTTP utilizado para activar la aplicación. Puede tomar los valores GET, POST, HEAD.
QUERY_STRING	Cadena con la información codificada que el cliente pasa a la aplicación CGI. Sólo se utiliza en el caso de accesos GET, ya que con POST toma un valor nulo.
CONTENT_TYPE	Tipo de datos que el cliente envía a la aplicación CGI. Normalmente, vale application/x-www-form-urlencoded, con las reglas de codificación antes descritas. Sólo se utiliza para los accesos POST.
CONTENT_LENGTH	Longitud de los datos que el cliente envía a la aplicación CGI. De esta forma, la aplicación sabe cuanta información debe leer de la entrada estándar. Sólo se utiliza para los accesos POST.
AUTH_TYPE	Tipo de autenticación HTTP empleada (basic, kerberos,...). En caso de no utilizarse, toma un valor nulo.
REMOTE_ADDR	Dirección IP del agente que envía la petición. En caso de acceso a través de proxy, no aparece la dirección del cliente, sino la del propio proxy.
REMOTE_HOST	Dirección DNS del cliente que envía la petición. En caso de accesos a través de un proxy, no aparece la dirección del cliente, sino la del propio proxy.
REMOTE_USER	Nombre del usuario que realiza la petición, en caso de que AUTH_TYPE sea basic.
PATH	Directorios por defecto de búsqueda de aplicaciones. Tiene el significado habitual dentro del sistema operativo.
PATH_INFO	Campo path-info de la URL de acceso a la aplicación CGI. Toma un valor nulo en caso de no utilizarse.
PATH_TRANSLATED	Directorio local resultante de sumar el directorio en donde residen los ficheros del servidor y PATH_INFO.
SCRIPT_NAME	URL local que identifica al programa CGI (p.e. /cgi-bin/prog.exe).
SERVER_NAME	Nombre DNS o dirección IP del servidor HTTP.
SERVER_PORT	Puerto TCP del servidor en que se recibe la petición.
SERVER_SOFTWARE	Tipo y versión del servidor HTTP que 'lanza' la aplicación CGI.
SERVER_PROTOCOL	Versión del protocolo HTTP que utiliza el servidor. Casi siempre es la 1.0.
GATEWAY_INTERFACE	Versión de protocolo que utiliza el servidor para dialogar con el programa. Normalmente es CGI/1.1

Variables de entorno accesibles en un CGI

tecnología web	
modelos básicos	
Contenido presentación introducción tecnología web middleware servicios web conclusiones	Modelos básicos <ul style="list-style-type: none"> Modelo Web Básico Modelo CGI (Common Gateway Interface) <pre> DOCUMENT_ROOT /usr/local/etc/httpd/htdocs GATEWAY_INTERFACE CGI/1.1 HTTP_ACCEPT image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */* HTTP_FORWARDED by http://cchp4.unican.es:8080 (Netscape-Proxy/1.12) HTTP_HOST cclxl HTTP_PRAGMA no-cache HTTP_PROXY_CONNECTION Keep-Alive HTTP_USER_AGENT Mozilla/3.0 (WinNT; I) PATH /sbin:/usr/bin:/bin:/usr/sbin:/etc:/etc QUERY_STRING REMOTE_ADDR 130.206.5.234 REMOTE_HOST cchp4.unican.es REQUEST_METHOD GET SCRIPT_NAME /cgi-bin/env.pl SERVER_NAME www.unican.es SERVER_PORT 80 SERVER_PROTOCOL HTTP/1.0 SERVER_SOFTWARE NCSA/1.5 </pre>

Método	Ruta-Aplicación-Parámetros	Versión	Cabecera
GET	/cgi-bin/buscar.cgi?nombre=Paco%20Maciá	HTTP/1.0	


```

<HTML>
<HEAD>
<TITLE>Búsqueda...</TITLE>
</HEAD>

<BODY>
<H1>Formulario de búsqueda</H1>
<HR>
<FORM method="GET" action="buscar.cgi">
  Introduzca un nombre:<br>
  <INPUT NAME="Nombre"><P>
  <INPUT type="submit" value="buscar"><P>
</FORM>
<HR>
</BODY>
</HTML>
          
```

Variable de Entorno	Valor
REQUEST_METHOD	GET
QUERY_STRING	/cgi-bin/buscar.cgi?nombre=Paco%20Maciá
CONTENT_LENGTH	39
SCRIPT_NAME	/cgi-bin/buscar.cgi
SERVER_PORT	80

Formulario HTML con invocación a CGI

Sistemas distribuidos

Contenido

- presentación
- introducción
- tecnología web**
- middleware
- servicios
- conclusiones

tecnología web

modelos básicos

Modelos básicos

Modelo Web Básico

ventajas

- Ofrece capacidad de respuesta dinámica
- Libertad elección lenguaje programación

desventajas

- No existe relación entre programa CGI y servidor Web

→ No existe control sobre ejecución, resultado, ...

- Nueva instanciación por cada solicitud

→ Sobrecarga de recursos

Sistemas distribuidos

tecnología web

modelos básicos

Contenido

- HTTP -> Protocolo de transporte sin estado
- Mecanismos de mantenimiento de estado (concepto de **sesión**)
 - Del lado del servidor
 - Ficheros, BD's,
 - Sistema centralizado -> Sobrecarga
 - Del lado del cliente
 - Campos ocultos
 - Cookies

Sistemas distribuidos

tecnología web

modelos básicos

Contenido

Ejemplo de uso de una cookie

En la cabecera de una respuesta:

```
Set-Cookie: NOMBRE=VALOR; expires=FECHA; path=PATH; domain=NOMBRE_DOMINIO; secure
```

```
Set-Cookie: ID=789; path=/
```

```
Set-Cookie: NOMBRE=Juan; path=/'
```

En la cabecera de la siguiente solicitud:

```
Cookie: NOMBRE1=VALOR1; NOMBRE2=VALOR2; ...
```

```
Cookie: ID=789; NOMBRE=Juan
```

Ejemplo de formulario con campos ocultos

```
<FORM METHOD="POST" ACTION="altacliente.cgi">
  <INPUT TYPE="HIDDEN" NAME="id" VALUE="789">
  <INPUT NAME="nombre" TYPE="TEXT">
  <INPUT NAME="apellidos" TYPE="TEXT">
  <INPUT TYPE="SUBMIT" value="Enviar">
</FORM>
```

Sistemas distribuidos

tecnología web

ampliaciones

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

- Modelos básicos
 - Modelo Web Básico
 - Modelo CGI (Common Gateway Interface)
- Ampliaciones
 - Mayor participación en el mundo de los negocios
 - Necesidad de generación de contenidos dinámicos
 - Ampliación del concepto Web como mero escaparate o catálogo estático
 - Mejor gestión de recursos y mayor organización
 - Aparición de numerosas tecnologías

Sistemas distribuidos

tecnología web

aplicación web

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

- Modelos básicos
 - Modelo Web Básico
 - Modelo CGI (Common Gateway Interface)
- Ampliaciones
 - Aplicación Web:** Aplicación basada en arquitectura C/S con cliente → navegador, servidor → servidor web, utilizando protocolos Internet (TCP/IP) para su entendimiento
 - Introducción anárquica de nuevas tecnologías en la parte cliente y en la servidora
 - Navegador: DHTML, ActiveX, applets Java (tecnología de plug-ins)
 - Servidor: Páginas Activas, Extensiones, servlets, servidor de aplicaciones (EJB, COM+, CCM)

Sistemas distribuidos

tecnología web

ampliaciones en el cliente

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Modelos básicos
 - Modelo Web Básico
 - Modelo CGI (Common Gateway Interface)
- Ampliaciones
 - Ampliaciones en el Cliente
 - DHTML
 - Applets de Java
 - Componentes: ActiveX

Sistemas distribuidos

tecnología web

ampliaciones en el cliente

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicio
- conclus

Modelos básicos

- Modelo Web Básico

ampliaciones en el cliente

```

graph LR
    Client[Cliente ligero Navegador Web] -- HTTP --> Internet((Internet TCP/IP))
    Internet -- HTTP --> Server[Servidor Web]
    Server -- HTTP --> Client
    Server --- Knowledge[know-how]
    Server --- Pages[Pág. HTML]
          
```


Sistemas distribuidos

tecnología web

ampliaciones en el cliente

Contenido

presentación

introducción

tecnología

middleware

servicios

conclusiones

```

<html>
<head>
<title>Sumar</title>
<script type="text/javascript">
function Sumar()
{
    var a, b, c;


    a = parseInt(miFormulario.campoA.value);
    b = parseInt(miFormulario.campoB.value);

    c = a + b;
    alert(a + " + " + b + " = " + c);
}
</script>
</head>

<body>
<form action="javascript:Sumar();" name="miFormulario" id="miFormulario">
<input name="campoA" type="text" id="campoA">
+
<input name="campoB" type="text" id="campoB">
<input type="submit" name="Submit" value="Sumar">
</form>
</body>
</html>

```

Ejemplo de código Javascript



Sistemas distribuidos

tecnología web

ampliaciones en el cliente

Contenido

presentación

Modelos básicos

```

<HTML>
<HEAD>
<TITLE>Ejemplo de Applet</TITLE>
</HEAD>

<BODY background="fondo.jpg">

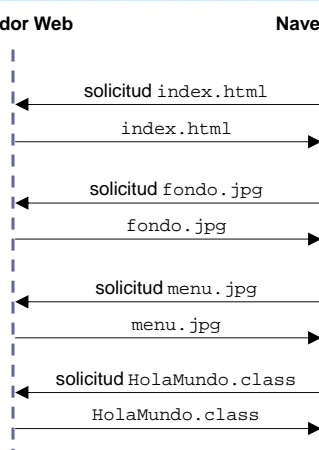
Esto es una imagen insertada:<BR>
<IMG src="menu.jpg">

Esto es un ejemplo de Applet:<BR>
<APPLET code = "appletHolaMundo.class"
width = "161"
height = "100" >
</APPLET>

</BODY>
</HTML>

```

Página index.html con referencias a dos imágenes JPG y un applet java



Secuencia solicitud-respuesta HTTP para completar referencias HTML con gráficos y applet java

Sistemas distribuidos

tecnología web

ampliaciones en el servidor

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Modelos básicos
 - Modelo Web Básico
 - Modelo CGI (Common Gateway Interface)
- Ampliaciones
 - Ampliaciones en el Cliente
 - DHTML
 - Applets de Java
 - Componentes: ActiveX
 - Ampliaciones en el Servidor
 - Modelo Servlets

Sistemas distribuidos

tecnología web

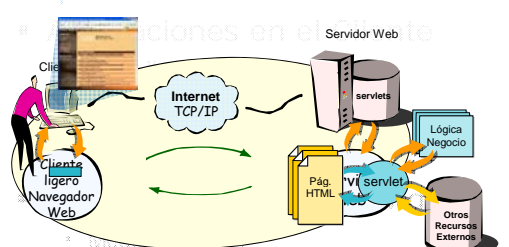
ampliaciones en el servidor

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Modelos básicos
 - Modelo Web Básico
 - Modelo CGI (Common Gateway Interface)
- Ampliaciones
 - Ampliaciones en el Cliente
 - Ampliaciones en el Servidor

modelo servlet



```

graph LR
    Client[Cliente ligero Navegador Web] -- "Internet TCP/IP" --> Server[Servidor Web]
    subgraph Server [Servidor Web]
        direction TB
        Servlets[servlets]
        Logic[Lógica Negocio]
        Resources[Otros Recursos Externos]
        HTML[Pág. HTML]
        Servlets --> Logic
        Logic --> Resources
        Logic --> HTML
        Resources --> HTML
    end
    HTML --> Client
          
```

Sistemas distribuidos

Contenido

presentar

introducción

tecnología

middleware

servicios

conclusiones

tecnología web

ampliaciones en el servidor

```

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class HolaMundoServlet extends HttpServlet {

    public void init(ServletConfig conf)
        throws ServletException
    {
        super.init(conf);
    } // de init

    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html");
        PrintWriter salida = res.getWriter();

        salida.println("<html>");
        salida.println("<body>");
        salida.println("<h1>Hola a Todos!</h1>");
        salida.println("</body>");
        salida.println("</html>");
    } // de service
} // de HolaMundoServlet

```

Código fuente del Servlet HolaMundo

Sistemas distribuidos

Contenido

presentar

introducción

tecnología

middleware

servicios

conclusiones

tecnología web

ampliaciones en el servidor

Características de los servlets

- Portabilidad. Sólo precisa motor JVM
- Rendimiento. Única instanciación
- Sesión. Mantiene información entre diferentes conexiones al servlet
- Software distribuido. Recuperación de disco/servidor de servlets. Comunicación entre sí
- Multithread. Múltiples peticiones concurrentemente

Sistemas distribuidos

tecnología web

ampliaciones en el servidor

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Modelos básicos
 - Modelo Web Básico
 - Modelo CGI (Common Gateway Interface)
- Ampliaciones
 - Ampliaciones en el Cliente
 - XHTML
 - Applets de Java
 - Componentes: ActiveX
 - Ampliaciones en el Servidor
 - Modelo Servlets
 - Modelo de Extensiones: ISAPI, NSAPI

Sistemas distribuidos

tecnología web

ampliaciones en el servidor

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

modelo de extensiones

• Modelo de Extensiones: ISAPI, NSAPI

Sistemas distribuidos

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

tecnología web

ampliaciones en el servidor

Modelos básicos

Modelo Web Básico

Common Gateway Interface

Ampliaciones en el Cliente

Componentes: ActiveX

Ampliaciones en el Servidor

Modelo Servlets

Modelo de Extensiones: ISAPI, NSAPI

Modelo de Páginas Activas: ASP, Liveware, JSP

Ventajas

Librerías de acceso dinámico

→ Sólo se instancian una vez

→ Se gana en rendimiento y aprovechamiento de los recursos del equipo

Desventajas

Tecnología cerrada a fabricante y servidor web: ISAPI/Microsoft/IIS o NSAPI/Netscape/NES

Sistemas distribuidos

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

tecnología web

ampliaciones en el servidor

Modelos básicos

Modelo Web Básico

Modelo CGI (Common Gateway Interface)

Ampliaciones

Ampliaciones en el Cliente

XHTML

Applets de Java

Componentes: ActiveX

Ampliaciones en el Servidor

Modelo Servlets

Modelo de Extensiones: ISAPI, NSAPI

Modelo de Páginas Activas: ASP, Liveware, JSP

22

Sistemas distribuidos

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

tecnología web

ampliaciones en el servidor

Modelos básicos

- Modelo Web Básico
- Modelo CGI (Common Gateway Interface)

modelo de páginas activas

Ampliaciones

Ampliaciones en el Servidor

- Modelo Servlets
- Modelo de Extensiones: ISAPI, NSAPI
- Modelo de Páginas Activas: ASP, Liveware, JSP

dis

Contenido

- presentación
- introducción
- Servlet equivalente
- conclusiones

JSP

holaMundo.jsp

```
<html>
<body>
<h2>
<%= "¡Hola a Todos!" %>
</h2>
</body>
</html>
```

SERVLET

equivalente

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class HolaMundoServlet extends HttpServlet {

    public void init(ServletConfig conf)
        throws ServletException {
        super.init(conf);
    } // de initServlet

    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter salida = res.getWriter();

        salida.println("<html>");
        salida.println("<body>");
        salida.println("<h2>¡Hola a Todos!</h2>");
        salida.println("</body>");
        salida.println("</html>");
    } // de service
} // de HolaMundoServlet
```

Ejemplo HolaMundo escrito como JSP y como Servlet

Sistemas distribuidos

tecnología web

ampliaciones en el servidor

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

Ventajas

- Habilita el concepto: sesión de usuario
- Buena gestión de recursos externos: DB
- Control sobre protocolo petición/respuesta
- Operaciones ejecutadas en servidor Web → Independiza aplicación Web de Navegador

Desventajas

- Necesidad de conocer lenguaje script
- Microsoft/ASP/VBScript
- Netscape/Liveware/Javascript
- Sun Microsystems/JSP/Java Script

Sistemas distribuidos

tecnología web

ampliaciones en el servidor

Contenido

presentación

introducción

tecnología web

middleware

servicios web

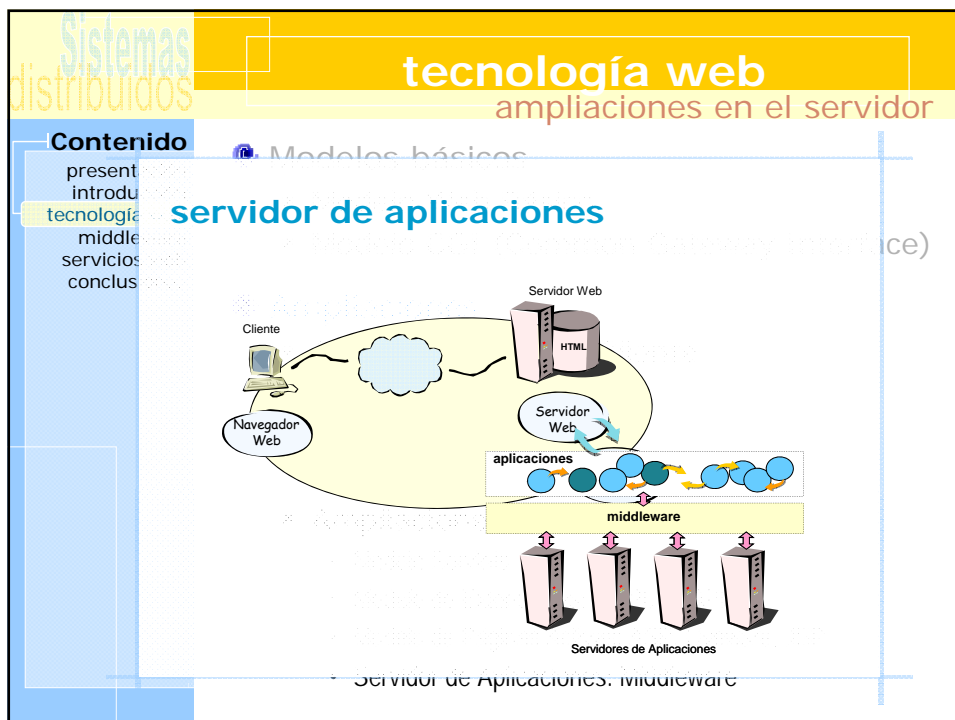
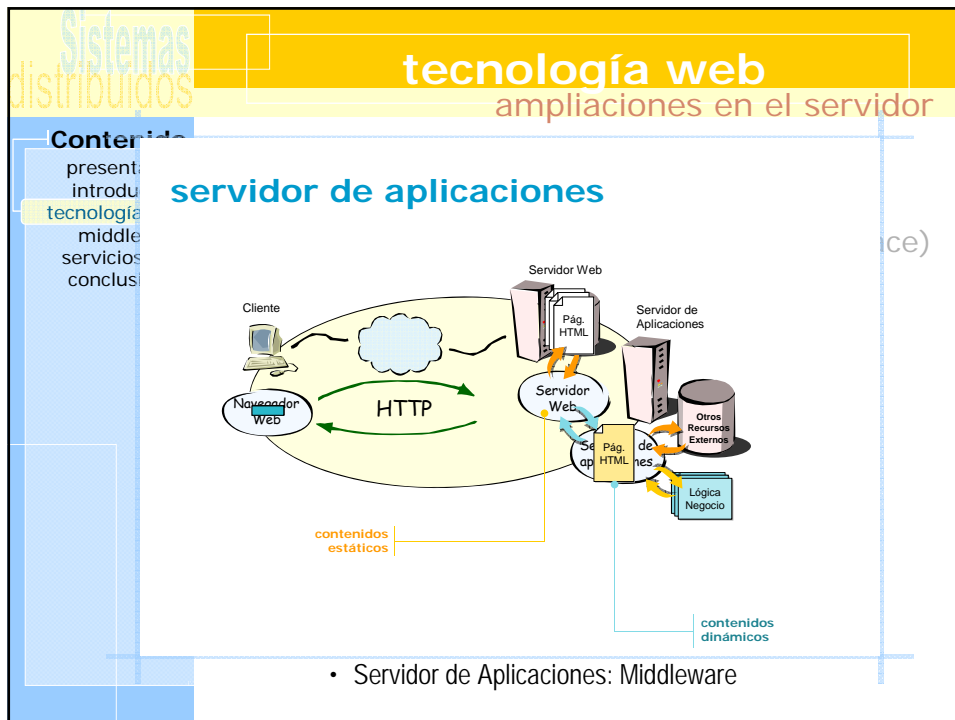
conclusiones

Modelos básicos

- Modelo Web Básico
- Modelo CGI (Common Gateway Interface)

Ampliaciones

- Ampliaciones en el Cliente
 - XHTML
 - Applets de Java
 - Componentes: ActiveX
- Ampliaciones en el Servidor
 - Modelo Servlets
 - Modelo de Extensiones: ISAPI, NSAPI
 - Modelo de Páginas Activas: ASP, Liveware, JSP
 - Servidor de Aplicaciones: Middleware



Sistemas distribuidos

tecnología web

middleware

Contenido
 presentación
 introducción
 tecnología web
 middleware
 servicios web
 conclusiones

servidor de aplicaciones

Modelos básicos

- Modelo CGI (Common Gateway Interface)

- Modelo de Extensiones: ISAPI, NSAPI
- Modelo de Páginas Activas: ASP, Liveware, JSP
- Servidor de Aplicaciones: Middleware

Sistemas distribuidos

tecnología web

modelo mixto

Contenido
 presentación
 introducción
 tecnología web
 middleware
 servicios web
 conclusiones

AJAX (Asynchronous Javascript And Xml)

- Tecnología de desarrollo Web
 - Conjunto de otras tecnologías existentes
 - HTML o XHTML, CSS, JavaScript, DOM, XML o JSON
- Aplicaciones ricas de la parte cliente
 - Aspecto similar al de aplicaciones standalone → evitar efectos refresco

Fundamentos

- Ejecución en el cliente (JavaScript)
- Conexión asíncrona con el servidor (XML)
 - Comunicación en segundo plano
 - Objeto XMLHttpRequest

Sistemas distribuidos

middleware

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

middleware

Sistemas distribuidos

middleware

definición

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Nivel de abstracción en términos de servicios cuya finalidad es proporcionar una visión única del sistema, independiente de la infraestructura que lo forme
- Abstrae de la heterogeneidad y complejidad de las redes de comunicaciones, sistemas operativos y lenguajes de programación
- p.v. de la arquitectura: modelo de servicios
- p.v. del programador: modelo de programación

Sistemas distribuidos

middleware

pv de la arquitectura

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Aplicaciones en términos de componentes
- Núcleo (elementos fundamentales)
 - Modelo de representación de datos
 - Modelo de comunicación
 - Servicios: seguridad, nombres, eventos y notificaciones, ...

```

graph TD
    subgraph Apps [aplicaciones (componentes)]
        direction TB
        A1[invocación a métodos remotos]
        A2[protocolo solicitud-respuesta]
        A3[representación externa de datos]
    end
    subgraph Core [Servicios middleware]
        direction TB
        C1[Notificaciones]
        C2[Seguridad]
        C3[Nombres]
    end
    A1 <--> C1
    A2 <--> C2
    A3 <--> C3
  
```

Sistemas distribuidos

middleware

pv de la arquitectura

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Aplicaciones en términos de componentes
- Núcleo (elementos fundamentales)
 - Modelo de representación de datos
 - Modelo de comunicación
 - Servicios: seguridad, nombres, eventos y notificaciones, ...
- Arquitectura comunicación
 - Interfaz
 - Proxy y réplica
 - núcleo

Sistemas distribuidos

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

middleware

pv de la arquitectura

Aplicaciones en términos de componentes

```

graph LR
    AC((Aplicación Cliente)) --> OP((Objeto Proxy))
    OP --> NSM1[núcleo servicios middleware]
    NSM1 <--> PC([Protocolo comunicación])
    PC <--> NSM2[núcleo servicios middleware]
    NSM2 --> RA((Réplica auxiliar))
    RA --> IO((Interfaz Objeto))
    IO --- C((Componente))
  
```

- Proxy y réplica
- núcleo

Sistemas distribuidos

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

middleware

pv del programador

Estructura de componente

- Interfaz: métodos + atributos
- Estado: estructura de datos, ...
- Comportamiento: lógica de aplicación

Acceso a los servicios

- API

```

graph LR
    C((cliente)) --> I[Interfaz  
- métodos  
- propiedades]
    subgraph componente
        I --- LN[Lógica Negocio]
        I --- E[(Estado)]
    end
  
```

API de Servicios Middleware

Sistemas distribuidos

middleware

modelo de componentes

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

- Reutilización de código
- Transparencia
 - con respecto a la plataforma sobre la que se ejecutan
 - con respecto al lenguaje de programación
- Capacidad de personalización a través de las propiedades
- Comunicación transparente entre ellos y con el contexto mediante eventos
- TIPOS componentes
 - Cliente
 - Servidor: Encapsulado de servicios o Almacén de datos

Sistemas distribuidos

middleware

j2ee

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

- Plataforma Middleware basada en Java
- J2EE ofrece 3 tecnologías para el desarrollo de componentes:
 - Componentes Web
 - Responden a una solicitud HTTP
 - Servlets**: programas Java que componen páginas web
 - JSP**: páginas web que contiene código Java
 - Componentes Enterprise JavaBean (**EJB**)
 - Modelo de componentes distribuidos
 - Unidades de software reusables que contienen lógica de empresa
 - Tipos:
 - Beans de sesión
 - Ejecuta solicitudes del cliente y es destruido cuando las operaciones del cliente se completan
 - Con estado: mantienen datos entre solicitud y solicitud
 - Sin estado: no mantienen estado entre solicitudes
 - Beans de entidad
 - Objeto persistente que modela los datos de un almacén
 - Beans dirigidos por mensaje
 - No son invocados por el cliente
 - Procesan mensajes asíncronos

Sistemas distribuidos

middleware

j2ee: api

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

APIs plataforma J2EE de SUN

- **RMI** Proporciona un mecanismo de acceso a objetos remotos como si fueran locales. Con RMI-JRMP solamente se podían instanciar remotamente objetos java. Con RMI-IIOP se pueden instanciar objetos CORBA.
- **JCA** Arquitectura que para interactuar con una variedad de EIS, incluye ERP, CRM y otra serie de sistemas heredados
- **JDBC** Acceso a base de datos relacionales
- **JTA** Manejo y la coordinación de transacciones a través de EIS heterogéneos
- **JNDI** Acceso a información en servicios de directorio y servicios de nombres
- **JMS** Envío y recepción de mensajes
- **JMail** Envío y recepción de correo
- **JIDL** Mecanismo para interactuar con servicios CORBA
- Otros APIs: tratamiento de XML, integración con sistemas heredados utilizando Servicios Web, ...

Sistemas distribuidos

middleware

j2ee: contendores

Contenido

presentación

introducción

tecnología web

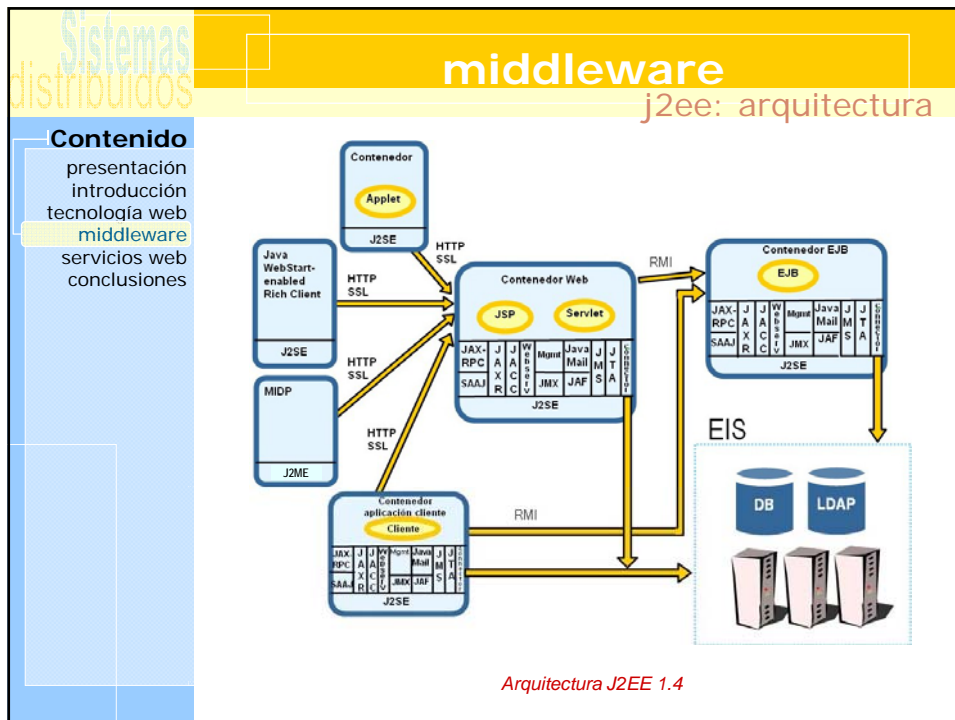
middleware

servicios web

conclusiones

J2EE establece 4 tipos diferentes de contendores:

- **Contenedor Web**
 - Servlets
 - JSP
- **Contenedor EJB**
 - Componentes Enterprise JavaBean
- **Contenedor Applet**
 - Applets: aplicaciones java ejecutadas en navegadores
- **Contenedor de aplicaciones cliente**
 - Aplicaciones Java estándar



Sistemas distribuidos

middleware

j2ee vs .net

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

	J2EE	.NET
Componentes	EJB	.NET Component
Mensajería	JMS	MSMQ
Transacciones	JTA	MTS
Acceso Distribuido	RMI RMI-IIOP	.NET Remoting
Conectores	JCA	Host Integration Server, Biztalk Server
Registro de Componentes	JNDI	Active Directory
Bases de Datos Relacionales	JDBC	ADO.NET
Manejo XML	JAXP	MSXML

Sistemas distribuidos

middleware

implementaciones

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Plataforma .NET
 - MS .NET Framework y .NET Remoting
 - Propietario de Microsoft
 - Es el único proveedor
- Plataforma J2EE
 - Java EE sdk
 - Aplicación de referencia
 - No eficaz
 - Oracle Weblogic
 - Oracle application server
 - Iplanet
 - IBM Websphere
 - Jboss (+ Tomcat), Geronimo, Glassfish

Sistemas distribuidos

integración

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

servicios web

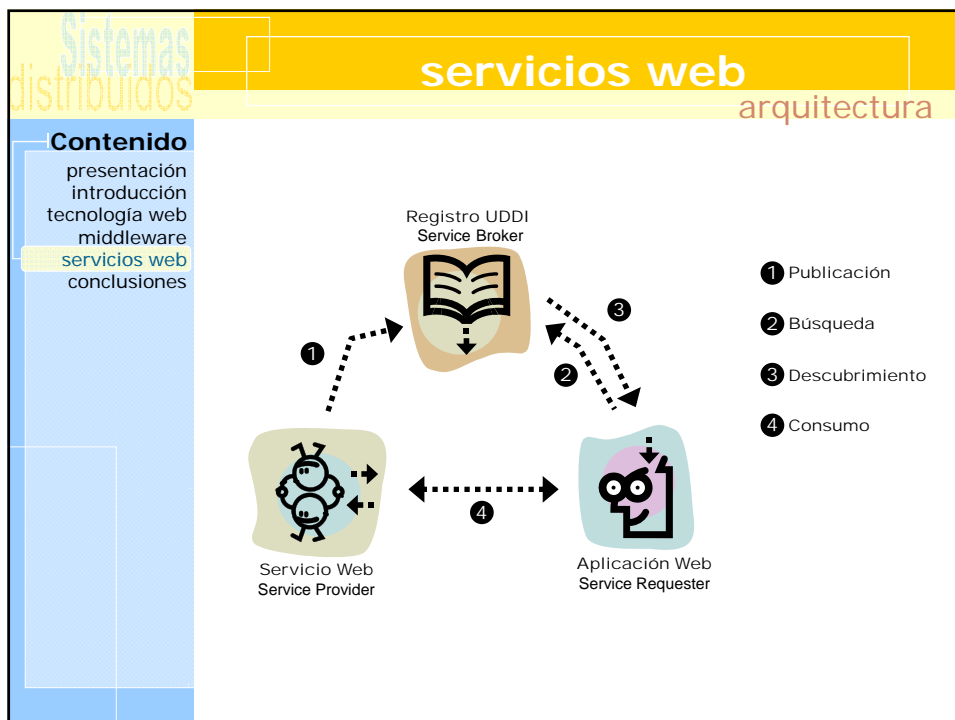
Sistemas distribuidos

servicios web

definición

Contenido
 presentación
 introducción
 tecnología web
 middleware
servicios web
 conclusiones

- Componentes que ejecutan procesos o funciones de negocio significativas, con una interfaz claramente definida y accesible a través de Internet, basada en el intercambio de documentos electrónicos en formato XML, y que pueden ser combinados entre sí.
- Paradigma para el desarrollo de sistemas distribuidos B2B
- Proporciona sistemas débilmente acoplados
- Reutilización y composición
- Interoperabilidad basada en contratos bien definidos
- Comunicación entre los negocios a nivel de aplicaciones
 - Utilización de estándares abiertos: XML, SOAP, UDDI y WSDL



Sistemas distribuidos	servicios web		protocolos
Contenido			
presentación			
introducción			
tecnología web			
middleware			
servicios web			
conclusiones			
	<ul style="list-style-type: none"> XML (eXtensible Markup Language) <ul style="list-style-type: none"> Solución para el intercambio de datos de una forma transparente SOAP (Simple Object Access Protocol) <ul style="list-style-type: none"> Plataforma para el intercambio de servicios sobre la red. Protocolo con el que hablan los servicios. Permite a las aplicaciones invocar métodos de objetos remotos Actúa sobre HTTP (en la práctica: sustituye a HTTP) UDDI (Universal Description, Discovery and Integration) <ul style="list-style-type: none"> Permite descubrir con quién comunicarse y dónde WSDL (Web Service Description Language) <ul style="list-style-type: none"> Describe los mensajes SOAP que definen un servicio Web en particular Utiliza un IDL (Interface Definition Language) de servicios Los registros UDDI <i>apuntan</i> a una hoja WSDL 		

Sistemas distribuidos	servicios web		wSDL
Contenido			
presentación			
introducción			
tecnología web			
middleware			
servicios web			
conclusiones			
	<ul style="list-style-type: none"> Documento XML para describir Servicios Web <ul style="list-style-type: none"> Descripción abstracta del servicio → Interfaz pública Descripción del protocolo a utilizar <ul style="list-style-type: none"> Las anteriores pueden ser común a varias implementaciones de Servicios Web Descripción de localización del servicio (implementación concreta) IDL para los servicios web W3C → WSDL 2.0 Anatomía de un documento WSDL <ul style="list-style-type: none"> <definitions>, contiene la definición de uno o más servicios. Secciones conceptuales: <message> y <portType>, qué operaciones provee el servicio <binding>, cómo se invocan las operaciones <service>, dónde se ubica el servicio <documentation>, puede contener información del servicio para el usuario Herramientas para generar la descripción de servicio WSDL <ul style="list-style-type: none"> WSTK de IBM .NET Studio de Microsoft AXIS gSOAP y cSOAP 		

Contenido

presentación
introducción
tecnología web
middleware
servicios web
conclusiones

Estructura de WSDL

Definición WSDL

Types → proveen definiciones de los tipos de datos utilizados para describir los mensajes intercambiados.

Message → representa una definición abstracta de los datos que están siendo transmitidos. Un mensaje se divide en una serie de partes lógicas, cada una de las cuales se asocia con alguna definición de algún sistema de tipos.

PortType → conjunto de operaciones abstractas. Cada operación hace referencia a un mensaje de entrada y uno de salida.

Binding → especifica un protocolo concreto y las especificaciones del formato de los datos de las operaciones y los mensajes definidos por un *portType* en concreto.

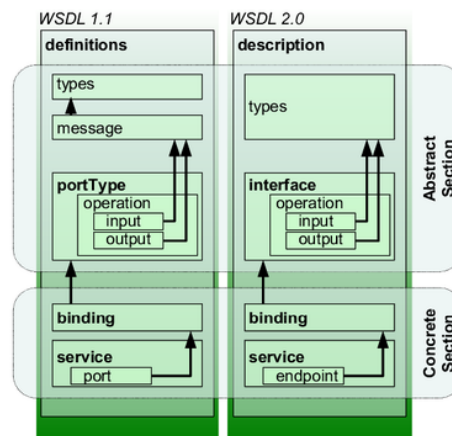
Service → para unir un conjunto de puertos relacionados.

Port → especifica una dirección para un *binding*, para así definir un único nodo de comunicación.

estructura wsdl

Contenido

presentación
introducción
tecnología web
middleware
servicios web
conclusiones



Sistemas distribuidos

Contenido

presentación

WSDL 1.1

```

<?xml version="1.0"?>
<definitions name="StockQuote"
targetNamespace="http://example.com/stockquote.wsdl"
xmlns:tns="http://example.com/stockquote.wsdl"
xmlns:xsd1="http://example.com/stockquote.xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/">

  <types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
      xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="TradePriceRequest">
        <complexType>
          <all> <element name="tickerSymbol" type="string"/> </all>
        </complexType>
      </element>
      <element name="TradePrice">
        <complexType>
          <all> <element name="price" type="float"/> </all>
        </complexType>
      </element>
    </schema>
  </types>

  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
  </message>
  <message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
  </message>

  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>

```

Sistemas distribuidos

Contenido

presentación

WSDL 1.1

estructura wsdl

WSDL 2.0

```

<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>

```

Sistemas distribuidos

servicios web

wsdl

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

WSDL binding: parámetros que influyen sobre la estructura del mensaje SOAP

- Style
 - RPC vs Document
- Use
 - Encoded vs Literal
- Combinación de los dos atributos → 4 posibilidades
- Document/literal wrapped

Trabajo → Analizar los 5 modelos

- WS-I
- ¿Cómo afecta a la generación del stub que enviará los mensajes SOAP?

Sistemas distribuidos

servicios web

wsdl

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

Ejemplo RPC/literal

```

<message name="miMetodoRequest">
  <part name="x" type="xsd:int" />
  <part name="y" type="xsd:float" />
</message>
<message name="empty" />
<portType name="pt">
  <operation name="miMetodo">
    <input message="miMetodoRequest" />
    <output message="empty" />
  </operation>
</portType>

```

El mensaje SOAP en RPC/literal sería:

```

<soap:envelope>
  <soap:body>
    <miMetodo>
      <x>3</x>
      <y>3.3</y>
    </miMetodo>
  </soap:body>
</soap:envelope>

```

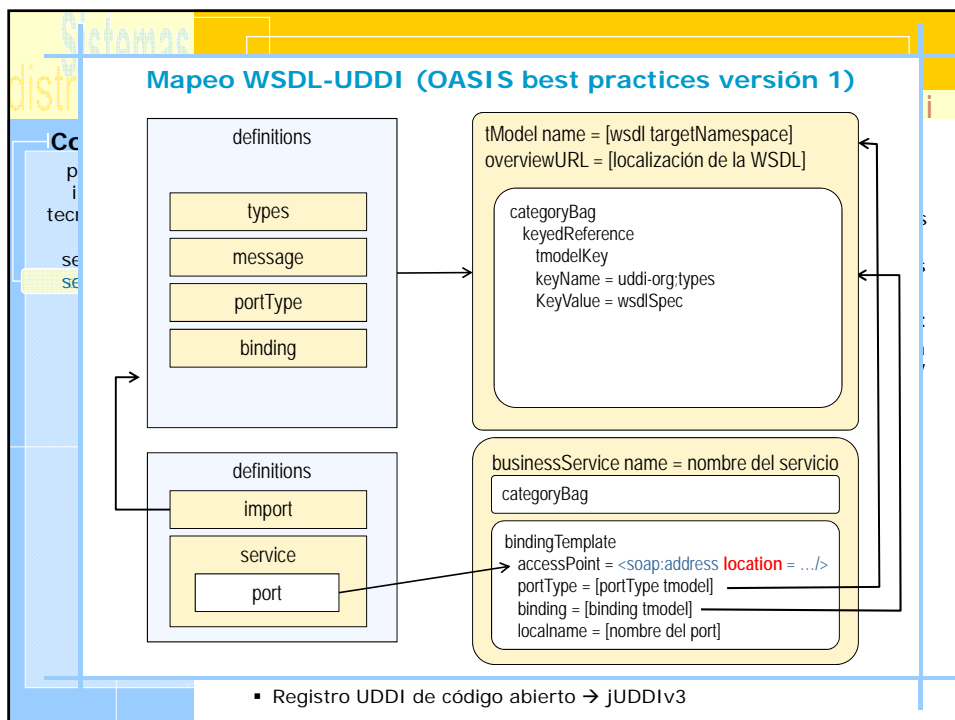
38

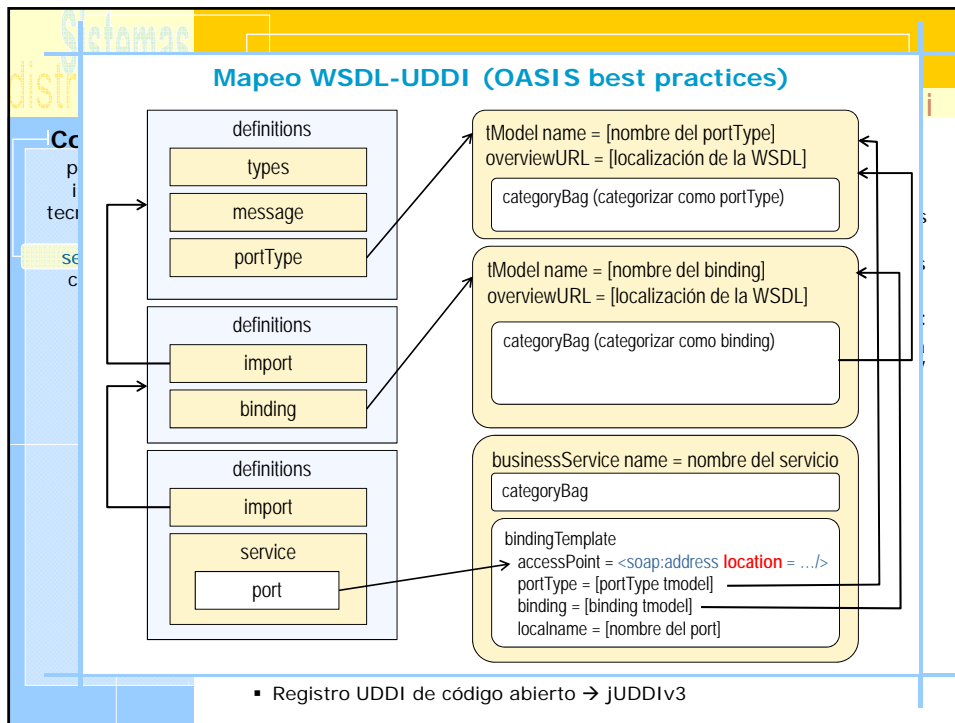
Sistemas distribuidos
servicios web
uddi

Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

- Secciones conceptuales de un registro UDDI:
 - Blanca** → Similar a la información que aparece en el directorio telefónico, que incluye nombre, teléfono, y dirección
 - Amarilla** → Similar a su equivalente telefónico, e incluyen categorías de catalogación industrial tradicionales, ubicación geográfica, etc.
 - Verde** → información técnica acerca de los servicios ofrecidos por los negocios
- Estructura central de un registro UDDI → 4 tipos de estructuras:
 - businessEntity** → Información sobre un negocio o entidad. Utilizada por el negocio para publicar información descriptiva sobre si mismo y los servicios que ofrece
 - businessService** → Servicios o procesos de negocios que provee la estructura businessEntity
 - bindingTemplate**: Datos importantes que describen las características técnicas de la implementación del servicio ofrecido
 - tModel** → Especificación y categorización técnica
- Características de UDDI. Dos categorías de API:
 - De **publicación** → Mecanismo para que los proveedores de servicios se registren (ellos mismos y sus servicios) en el Registro UDDI.
 - De **consulta** → Permite a los subscriptores **buscar** los servicios disponibles y **obtener** el servicio una vez localizado
- <http://www.uddi.org> → Microsoft e IBM
- Registro UDDI de código abierto → jUDDIv3





Sistemas distribuidos

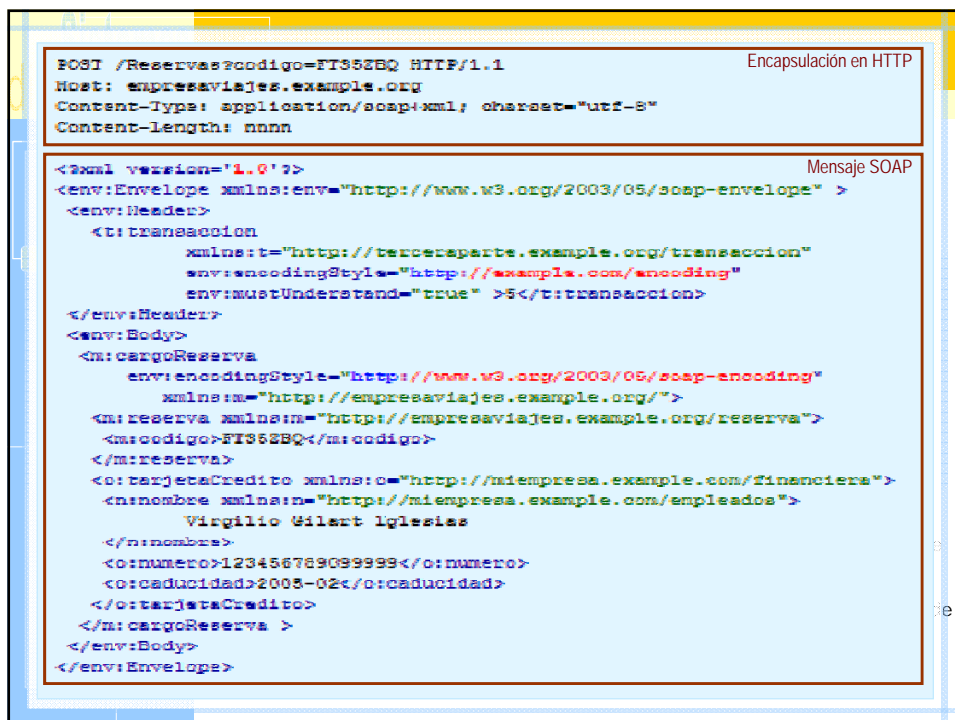
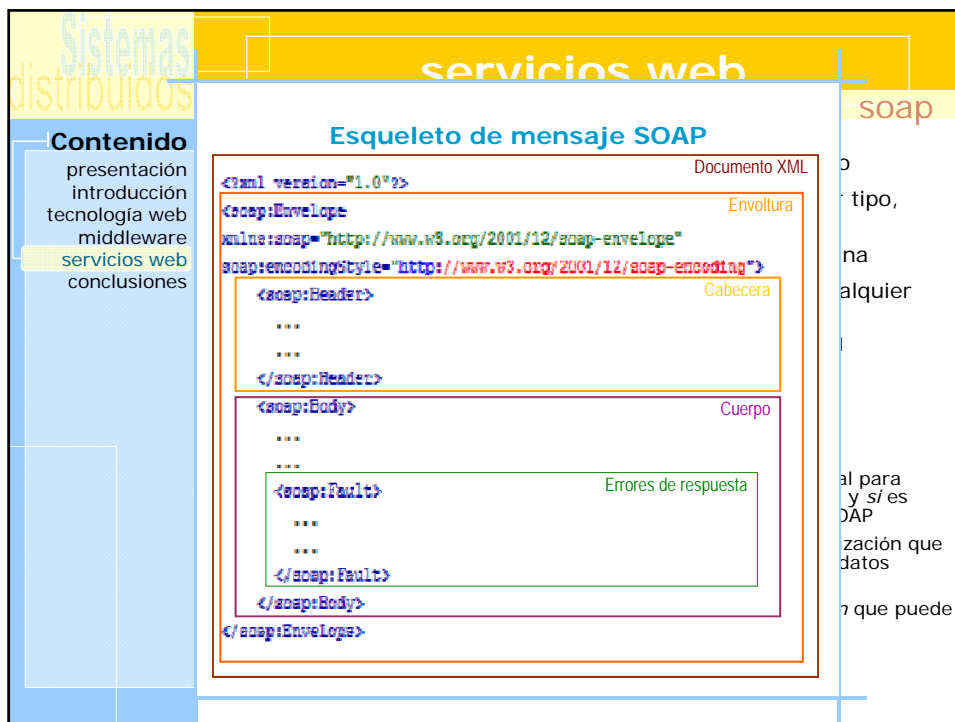
Contenido

- presentación
- introducción
- tecnología web
- middleware
- servicios web
- conclusiones

servicios web

soap

- Protocolo ligero basado en XML para el intercambio de información en un entorno descentralizado y distribuido
- Facilita la intercomunicación entre objetos de cualquier tipo, sobre cualquier plataforma, en cualquier lenguaje
- Comunicación (débilmente acoplada) máquina-a-máquina
- Una petición/respuesta de SOAP puede viajar sobre cualquier protocolo de aplicación: HTTP, SMTP, ...
 - W3C en la especificación determina solo HTTP como oficial
- Protocolo extensible desarrollado por el W3C
- Permite el intercambio a través de firewalls
- **Mensajes SOAP (XML):**
 - **Sobre ó envoltura:** define un marco de referencia general para expresar *qué* hay en el mensaje, *quién* debe de atenderlo, y *si* es opcional u obligatorio. Identifica un mensaje XML como SOAP
 - **Reglas de codificación:** definen un mecanismo de serialización que se puede utilizar para intercambiar instancias de tipos de datos definidos por la aplicación
 - **Representación RPC/Document:** define una *convención* que puede ser utilizada para representar las llamadas y respuestas a procedimientos remotos



Sistemas

Encapsulación en SMTP

De: a.ovvind@miempresa.example.com

A: reservas@empresaviajes.example.org

Asunto: Viaje a LA

Fecha: Thu, 29 Nov 2001 13:20:00 EST

Message-Id: <EE492E162080090276D204424960C0C@miempresa.example.com>

Content-Type: application/soap+xml

Mensaje SOAP

```

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    ...
  </env:Header>
  <env:Body>
    <p:itinerario
      xmlns:p="http://empresaviajes.example.org/reserva/viaje">
      <p:ida>
        ...
      </p:ida>
      <p:vuelta>
        ...
      </p:vuelta>
    </p:itinerario>
    <q:alojamiento
      ...
    </q:alojamiento>
  </env:Body>
</env:Envelope>

```

Sistemas distribuidos

servicios web

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

conclusiones

Sistemas distribuidos

Contenido

presentación

introducción

tecnología web

middleware

servicios web

conclusiones

conclusiones

- Gran diversidad de tecnologías alrededor de la Web
- Idoneidad del modelo de componentes y plataforma middleware para el desarrollo de grandes aplicaciones distribuidas sobre Internet
- Idoneidad del modelo web para gestionar la capa de presentación (B2C)
- Necesidad de conocer tecnologías y herramientas para seleccionar adecuadamente
- Idoneidad del modelo de servicios Web para gestionar la interacción B2B

introducción

al modelo de

Tecnologías Web y

Middleware

middleware

CORBA

J2EE

© 2005 por

frupen