

# Evolución de los Sistemas Distribuidos

Universidad EAFIT  
Departamento de Informática y Sistemas  
Sistemas Distribuidos

## Contenido

- Evolución de las Aplicaciones Distribuidas
- Soluciones propuestas para Interoperabilidad
- Modelo de Aplicaciones Web
- Que son los Servicios Web
- Cuales son los protocolos y tecnologías relacionados con los Servicios Web
- Algunas aplicaciones claves de los Servicios Web
- Conclusiones

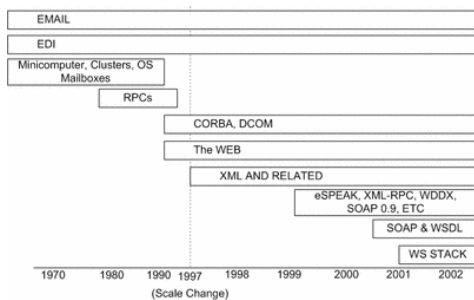
## Antecedentes (1/2)

- La idea general alrededor de Web Services no es algo nuevo.
- Antecesoros en el tema de computación distribuida han existido y funcionado:  
RPC, EDI, CORBA, COM, APPC
- La diferencia se encuentra en el alto nivel de encapsulación e independencia entre las aplicaciones.
- El gestor de la iniciativa es W3C, garante de evolución y futuro.
- "Programa a usuario" → Semantic Web

## Antecedentes (2/2)

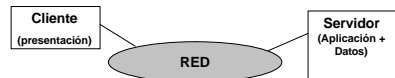
- Evolución tecnológica:  
Código libre  
Programación estructurada  
Programación orientada a objetos  
Objetos distribuidos - Componentes
- Evolución de integración de aplicaciones:  
Computación por departamentos  
Interfaces punto a punto ("spagueti code")  
Sistemas ERP  
Soluciones de "middleware"
- Necesidad de establecer conexiones tanto entre aplicaciones internas como B2B.

## HISTORY OF DISTRIBUTED COMPUTING



## Evolución de las Aplicaciones Distribuidas (1)

- Sistema Distribuido: "Colección de máquinas/procesos que colaborar para cumplir un objetivo"
- Inicio con Aplicaciones Centralizadas. Todo lo hacia un mismo equipo.
- Primer servicio telemático: Emulación de Terminal
  - Hay distribución, pero todo lo sigue haciendo el Servidor.
  - Este modelo no es malo, como se critico en su momento, de hecho el mismo Web simula esta situación
  - Ej: telnet, Xwindows, Windows Terminal, VNC, etc.



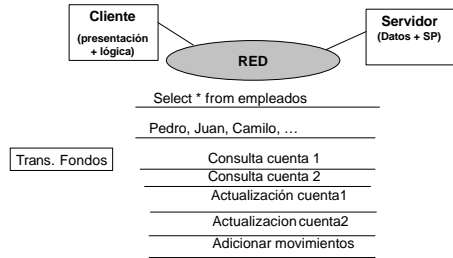
## Evolución de las Aplicaciones Distribuidas (2)

### Cliente/Servidor con Bases de Datos

- Modelo de 2 niveles
- Aparición de n-lógicas
  - Presentación
  - Comunicaciones
  - Lógica del Negocio
  - Datos
- En el Cliente se haya la Presentación y la Lógica del Negocio
- En el Servidor se hayan los Datos (Bases de Datos)
- Se supone que las entidades intercambian sentencias SQL
- NO orientado a transacciones
- Muy orientado a 4GL
- Procedimientos almacenados. Lógica del negocio en la base de datos. Dependiente.

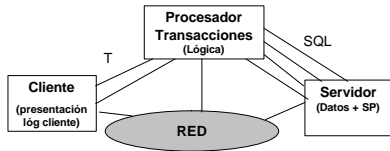
## Evolución de las Aplicaciones Distribuidas (3)

- Cliente/Servidor con bases de datos



## Evolución de las Aplicaciones Distribuidas (4)

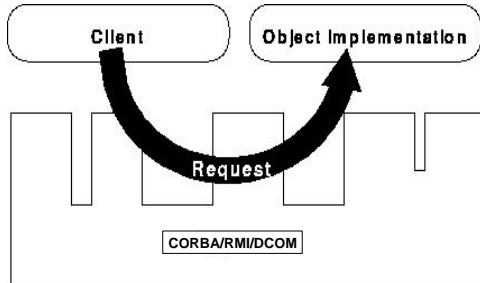
- Procesadores de Transacciones:
  - Orientado a transacciones
  - 3 niveles
  - OLTP



## Objetos Distribuidos

- Primeros pasos en RPC's
- A finales de los 80's emergió DCE (Distributed Computing Environment) como una iniciativa para estandarizar las diferentes tecnologías de RPC. No considera tecnologías de Mensajería.
- Éxito del modelo Orientado a Objetos tanto en Análisis/Diseño como en Desarrollo.
- Por qué no extender este modelo a un ambiente distribuido
- Un cliente en cualquier parte de la red, invoca un método de un objeto remoto.
- Adecuados en comunicaciones:
  - Cliente a Servidor
  - Servidor a Servidor

## Objetos Distribuidos



## Objetos Distribuidos

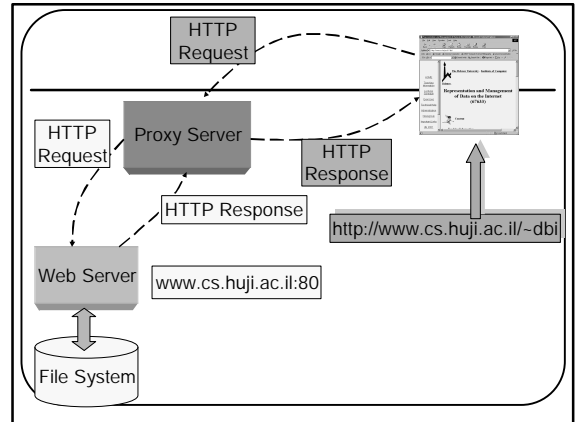
### Arquitecturas propuestas

- RMI (Invocación Remota de Métodos)
  - Independiente del sistema operativo
  - Solo un lenguaje: JAVA
- DCOM de Microsoft
  - Un solo sistema operativo: Familia Microsoft
  - Varios Lenguajes: Visual Basic, C++, C
- CORBA de la OMG
  - Independiente del Sistema Operativo
  - Varios lenguajes: Java, c, c++, ada, cobol, vb, ...



## World Wide Web

- Llegada del Web
  - Diseñado originalmente para acceder a recursos compartidos
  - Protocolo de comunicación HTTP
    - Protocolo sin estado (stateless)
    - Browser – Servidor Web: GET recursos
    - Servidor Web – Browser: recurso estático (html, imágenes, ...)
  - Formateo de datos
    - HTML, sintaxis y semántica de visualización
  - Diferentes tipos de recursos
    - MIME (texto, html, imágenes, binarios, videos, etc)
  - Navegación y localización NO lineal
    - Hipertexto
    - Hipermedia



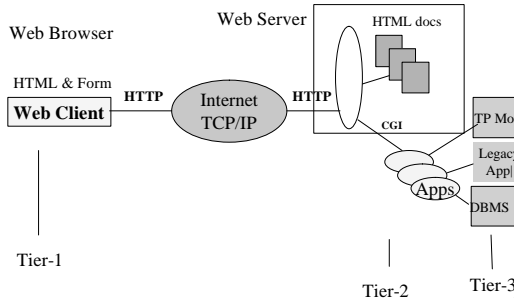
## Web como plataforma para Aplicaciones Distribuidas

- Aplicaciones Web (Web Applications)
- HTTP/HTML fue diseñado originalmente para acceder a recursos compartidos, NO para ejecución de aplicaciones remotas
  - Recursos estáticos
- Porqué no permitir solicitar datos al usuario y ejecutar aplicaciones vía web?
  - Como capturar entrada del usuario?
    - FORMAS: <FORM ...> <INPUT...> ... </INPUT> ... </FORM>
  - El recurso es el nombre de una aplicación

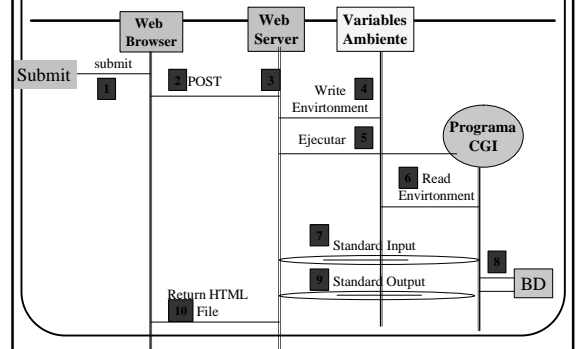
## HTTP/CGI

- Web Form es una página HTML con uno o más campos de entrada y un boton obligatorio "Submit"
- Una vez "click" sobre el boton "Submit" el browser envia el contenido de los campos de entrada hacia el servidor.
- Arma un mensaje HTTP para enviar el requerimiento y luego invoke un método GET o POST en el lado del servidor.
- El servidor pasa el requerimiento y sus parametros a otro programa usando un protocolo llamado CGI

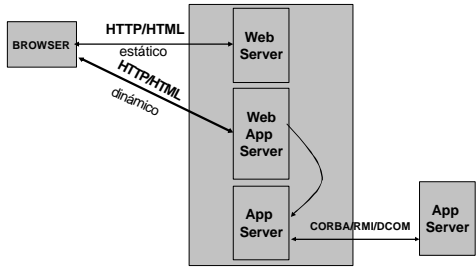
## 3-Tier Client/Server, Web-Style



## Ejecución programa CGI



## Integración de Aplicaciones Web con Objetos Distribuidos



## Componentes Distribuidos

- Evolución de los Objetos Distribuidos
- Bloques de construcción de aplicaciones llamados COMPONENTES
- Conceptos de CONTENEDORES
- Servicios disponibles para los Componentes ofrecidos por el Contenedor
  - Comunicaciones
  - Transacciones
  - Ejecución
  - Ciclo de Vida
  - Persistencia
  - ...
- Plataformas más representativas
  - J2EE, con JSP/Servlets como tecnologías de Web Application
  - DNA, con ASP como tecnología Web Application

## Arquitectura J2EE

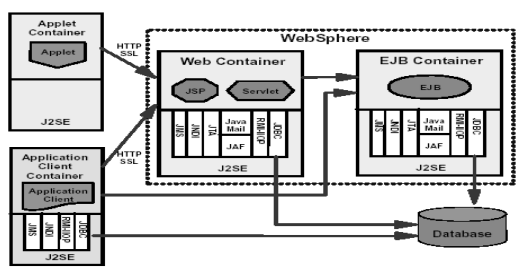
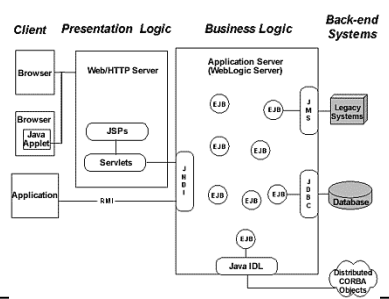
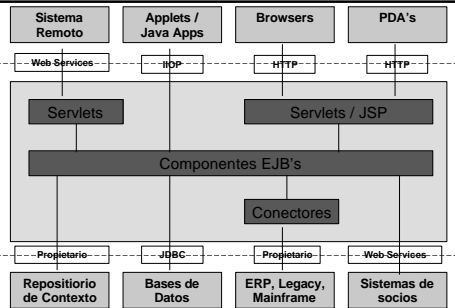


Figure 1-1 J2EE architecture diagram

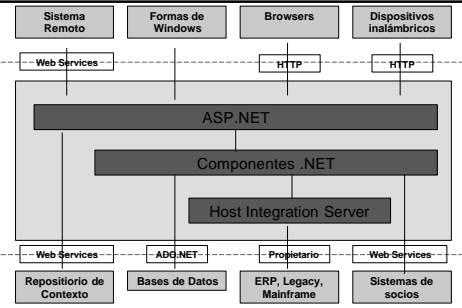
## Aplicación N-Niveles en Java



## ARQUITECTURAS DE COMPONENTES J2EE



## ARQUITECTURAS DE COMPONENTES Microsoft .NET



## INTRODUCCIÓN A WEB SERVICES

- *Los Web Services son una arquitectura de computación distribuida en evolución que usan sus propias interfaces programa-programa, protocolos y servicios de registro de tal manera que posibilitan que aplicaciones de diferentes plataformas tecnológicas puedan utilizar "servicios" de otras aplicaciones.*
- *Un Web Service se aprovecha de la especificación de XML para definir tanto su descripción, como los mensajes que recibe y produce al igual que en los servicios de registro del servicio.*

## Qué son los Web Services?

- Semánticamente encapsula funcionalidades discretas.
- Sistemas Debilmente-Acoplados, Componentes Reutilizables
- Accesibles programáticamente
- Distribuido Sobre Internet
- Dado que todas las comunicaciones se realizan en XML los Web Services no dependen de sistemas operativos específicos
- De acuerdo a esto, Java puede hablar con Perl, Windows con Unix.

## Cont.

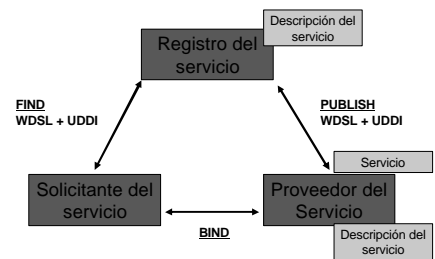
- De la definición anterior se desprende que los Web Services tienen dos propiedades adicionales:

1) Deben tener una interfaz pública definida en una gramática común en XML, la interfase describe todos los **MÉTODOS** disponibles a los clientes y especifica la **firma** para cada método

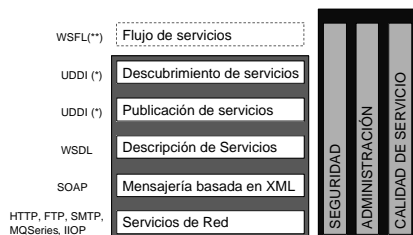
La definición de esta interfaz se hace con el lenguaje **WSDL (Web Service Description Language)**

2) Si usted crea un servicio Web, debe tener una forma de publicarlo, debe existir una forma de localizar el servicio y localizar su interfase pública, esto se hace con **UDDI (Universal Description, Discovery, and Integration)**

## INTRODUCCIÓN A WEB SERVICES Actores, Objetos y Operaciones



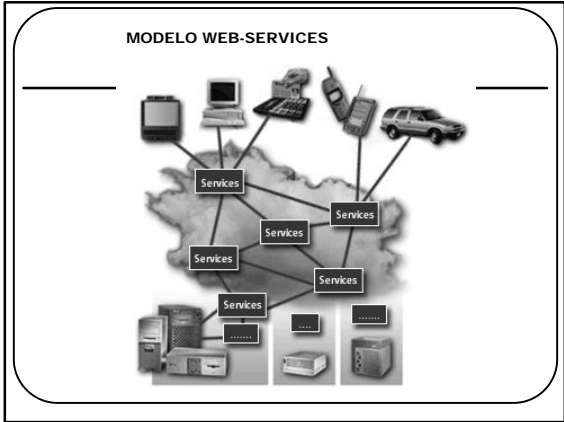
## INTRODUCCIÓN A WEB SERVICES Stack de Componentes



(\*) No es un estándar de la industria, ha sido propuesto por Microsoft e IBM entre otros.  
(\*\*) Es una propuesta de IBM

## Componentes de Web Services

- **UDDI:** "Universal Description, Discovery and Integration". Permite que las aplicaciones puedan ser listadas y localizadas de manera ágil y eficaz. (No es un estándar actualmente).
- **WDSL:** "Web Services Description Language". Es una plantilla o interfaz que permite a las aplicaciones el describirle a otras aplicaciones, las reglas para interactuar entre sí.
- **SOAP:** "Simple Object Access Protocol" Provee los mecanismos para la ejecución de llamadas a procedimientos remotos entre programas, de tal manera que se puedan establecer de manera eficiente las comunicaciones de tipo "programa a programa".



- ### Algunas de las Principales Aplicaciones de los Web Services
- Interoperabilidad entre ambientes de desarrollo
    - Tradicionalmente, antes de los Web Services una empresa debía decidir en que ambiente de desarrollo trabajar
      - "Me voy por JAVA"
      - "Me voy por Microsoft"
      - "Me voy por Oracle"
    - Una vez se decidía el Ambiente de Desarrollo, queda uno "Cazado" con dicha plataforma.
    - Con Web Services, desarrolle sus aplicaciones en lo que más le guste e intégralas con Web Services.
  - Una aplicación que parecía Irreal:
    - Llamar desde una aplicación hecha en Java y servicio hecho en ambiente Microsoft.
    - Esto es posible gracias a los Web Services.

- ### Ebussines y Agentes Inteligentes
- Antes de los Web Services
    - La mayoría de los sistemas de Comercio Electrónico (Customer to Business, C2B) se realizaron con APLICACIONES WEB
    - Las comunicaciones entre los servidores de aplicaciones involucrados eran Proprietarios y alguna de las partes colocaba las condiciones.
    - Ej: Los primeros pasos al Comercio Electrónico en Colombia, los Bancos determinaban los formatos, tecnologías, ... para comunicarse con ellos.
    - Todas las Interacciones a nivel de servicio son entre Cliente y Proveedor.

- ### Ebussines y Agentes Inteligentes
- Con los web services
    - Se instala en Internet, una Agencia de Viajes Virtual
    - Su función es principalmente como Broker
    - Deseo realizar un viaje, en el cual obviamente necesito:
      - Tiquetes Aereos
      - Hotel
      - Auto en la ciudad destino
    - Inicio una transacción en Internet para realizar mi viaje entre una fecha 1 y fecha 2 con algún nivel de holgura.
    - Dicha transacción es atendida por un Agente Inteligente de Software que realizará en mi nombre la transacción.
    - El Agente tiene toda la información para llevar a cabo la transacción así como el manejo de restricciones y flujo de procesos.

