



SISTEMAS DISTRIBUIDOS BASADOS EN LA WORLD WIDE WEB

Sistemas informáticos I



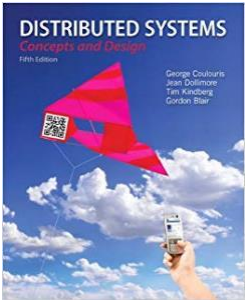
ÍNDICE

- Introducción a los Sistemas distribuidos basados en WWW
- Web hipertexto
 - HTTP
 - SGML
 - HTML
 - CSS
 - XML
 - JSON
- Web interactiva (Aplicaciones Web)
 - Formularios HTML
 - CGIs
 - Web APIs e interfaces híbridas
 - Páginas dinámicas
 - PHP y Python
- Ejecución de código en el cliente
- HTML5
- Aplicaciones Web
- Servidores de aplicaciones

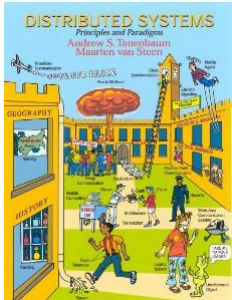


REFERENCIAS BIBLIOGRÁFICAS GENERALES

COULOURIS, G., DOLLIMORE, J. y
KINDBERG, T., Sistemas distribuidos.
Conceptos y diseño, Addison-Wesley



TANENBAUM, A. y VAN STEEN, M.,
Distributed Systems. Principles and
paradigms, Prentice Hall



REFERENCIAS BIBLIOGRÁFICAS ESPECÍFICAS

- W3C, mejor referencia casi para todo:
 - Manuales de referencia: <https://www.w3schools.com>
 - URI/URL: <http://www.w3.org/TR/uri-clarification/>
 - ...
- Miles de libros y referencias web
- Referencia de Python: <https://docs.python.org>
- Manual de PHP: <http://www.php.net/manual/es/>
- Sun Microsystems, The J2EE Tutorial
- Microsoft, Understanding Enterprise Platforms:
http://msdn.microsoft.com/netframework/default.aspx?pull=/library/en-us/dnpag/html/jdni_ch02.asp#jdni_ch02_topic3



2.1 INTRODUCCIÓN

Sistemas distribuidos basados en la World Wide Web



SISTEMAS DISTRIBUIDOS BASADOS EN LA WORLD WIDE WEB

- En origen, **sistemas distribuidos** basados en la extensión del modelo de **cliente ligero** bajo **protocolo HTTP** para el intercambio de información
- La generalización del uso de la WWW como mecanismo de intercambio de información produjo un cambio en el modo de diseñar las aplicaciones distribuidas
 - El **cliente Web** se convierte en la interfaz de uso genérico para la interacción de los usuarios con un sistema distribuido
 - En el modelo actual, las carencias del modelo de cliente ligero universal se suplen con elementos de programación complementarios
 - El **servidor Web** pasa a ser el núcleo básico donde se gestiona la ejecución de programas
 - Es necesario completar su funcionalidad inicial para:
 - Ejecutar programas (~ **servidor de Aplicaciones**)
 - Enlazar con programas ya existentes (**servidor de Back-end**)
- En este tema se tratan los distintos modos que se han ido desarrollando para implementar este modelo de sistemas hasta llegar al modelo actual



INTERNET, INTRANET, EXTRANET

- Los tres son términos que se refieren a redes de comunicación físicas que principalmente se diferencian por el tipo de acceso, los usuarios que tienen acceso a ella y el tipo de información que almacena:
 - **Internet:** Red pública, externa, comunicación con el mercado. Soporte *business to customer (b2c)*
 - **Intranet:** Red IP privada. Soporte de comunicaciones de un grupo de usuarios: empresa, institución... Soporte *business to employee (b2e)*
 - **Extranet:** Acceso a una Intranet por usuarios externos: Clientes, proveedores... Soporte *business to business (b2b)*
- A nivel de arquitectura básica, no existen diferencias entre los sistemas distribuidos diseñados para internet, una intranet o una extranet:
 - Sus diferencias provienen fundamentalmente de las distintas necesidades de seguridad



WORLD WIDE WEB

- Colección de **documentos**
- Cada documento puede contener objetos de diversos tipos:
 - Texto, gráficos, imágenes, voz, vídeo, otros documentos...
- Los objetos se reconocen mediante su *Uniform Resource Identifier*, URI
- Cada documento puede tener referencias a cualquier otro objeto en la red. Se denominan hiperenlaces
- El cliente accede a los documentos a través de un visualizador (*browser*) o navegador → cliente ligero
- Los nodos de la red presentan los documentos a través de un servidor Web
- Comunicación mediante protocolo HTTP (*HyperText Transfer Protocol*) y cada vez más HTTPS
- Contenido de las documentos descrito en lenguaje HTML (*HyperText Markup Language*). Actualmente HTML5



URI: UNIFORM RESOURCE IDENTIFIER

- URI, *Uniform Resource Identifier* (RFC 2396)
 - URN (*Uniform Resource Name*): identifica un recurso o espacio de nombres (*namespace*). Ejemplo: urn:isbn:n-nn-nnnnnn-n
 - URL (*Uniform Resource Locator*): indica dónde/cómo encontrar el recurso
- Extensión del nombre de un fichero que permite identificar recursos en Internet
 - *esquema:parte-dependiente-del-esquema*
 - La parte dependiente del esquema se basa en *namespaces* y *subnamespaces*
- En el caso de las URLs, contienen datos sobre:
 - El mecanismo de acceso primario (protocolo cómo se transmite la información)
 - La dirección del componente de la red que contiene el recurso
 - El identificador del recurso dentro del componente que lo contiene
- Dos tipos de URLs:
 - Absolutas
 - `http://www.webcon.com/~tbrown/computer.html`
 - Relativas
 - `img/br2.gif`



ESTRUCTURA DE UNA URL

- Esquema: asociado con alguno de los protocolos de comunicación o servicios disponibles en Internet:
 - ftp, http, gopher, mailto, news, file, telnet...
- Parte-dependiente-del-esquema: datos necesarios para localizar el objeto
 - En su mayoría, su *namespace* responde a la estructura:
`//[usuario[:clave]@]host[:puerto]/[path]`
 - El parámetro puerto se puede omitir y se tomaría el puerto por defecto para el protocolo utilizado
- Ejemplos de namespaces URL:
 - `ftp://[usuario[:contraseña]@]host[:puerto]/[path-del-archivo]`
 - `ftp://ftp.ibm.com/programs/download/testip.zip`
 - `http://[host][:puerto]/[path-recurso][?consulta]`
 - `http://search.yahoo.com/bin/search?p=url+uri`
 - `mailto:[Dirección-RFC822]`
 - `mailto:bobsponja@fondobikini.com`

