

# Caso de Estudio: World Wide Web

Virginia Padilla

Universidad Nacional Experimental de Guayana

*virginiapadillas@gmail.com*

8 de noviembre de 2024

# Contenido

- 1 World Wide Web
  - HTML
  - URL
  - HTTP
  - XML

## Caso de Estudio: World Wide Web

### Propósito

El caso de estudio que se presenta está tomado de [Coulouris, 2011].

El propósito es la presentación de las tecnologías fundamentales sobre la cual están construidos los sistemas distribuidos.

## Caso de Estudio: World Wide Web

### Características

- 1 La World Wide Web es un sistema abierto a los tipos de recursos.
- 2 Su funcionamiento se basa en estándares de comunicación y contenido que se publican e implementan libremente.
- 3 Tipos de navegador, en varias plataformas; así implementaciones de servidores web. Los usuarios tienen acceso a los navegadores desde los dispositivos que utilizan.
- 4 Los navegadores están diseñados para adaptarse a nuevas funcionalidades de presentación de contenido en forma de aplicaciones auxiliares y *plug-ins*
- 5 Los navegadores se adaptan a nuevos contenidos, protocolos, formatos.

# Tecnologías

Las tecnologías sobre la que se basa la Web son estándares de la *World Wide Web Consortium W3C*.

- Lenguaje de Hypertexto, *HTML*
- Lenguaje solicitador de recurss, *URL*
- Protocolo de solicitud-respuesta, *HTTP*
- Lenguaje de marcado de Hypertexto, *XML*

# HyperText Markup Language: HTML

El lenguaje de marcado de hipertexto, *HTML* se utiliza para especificar el texto e imágenes que componen el contenido de una página web, y cómo se colocan y formatean, para su presentación al usuario.

Una página web contiene dichos elementos estructurados como títulos, párrafos, tablas e imágenes. *HTML* también se utiliza para especificar enlaces y recursos que están asociados a ellos.

## HTML

Conozca más de URL visitando este sitio HTML

# HTML

Fragmento típico de texto *HTML* en figura 1

- El texto se almacena en un archivo al que accede un servidor web.
- Un navegador recupera el contenido de este archivo de un servidor web.
- El navegador lee el contenido devuelto por el servidor y lo convierte en texto formateado e imágenes distribuidas en una página web de una manera adecuada. Solo el navegador, no el servidor, interpreta el texto .

```
<IMG SRC = "http://www.cdk5.net/WebExample/Images/earth.jpg">      1  
<P>                                                                    2  
Welcome to Earth! Visitors may also be interested in taking a look at the 3  
<A HREF = "http://www.cdk5.net/WebExample/moon.html">Moon</A>.      4  
</P>                                                                    5
```

# HTML

- El servidor informa al navegador del tipo de contenido que está devolviendo, para distinguirlo de, por ejemplo, un documento en formato de documento portátil.
- El servidor puede inferir el contenido escrito cuando reconoce la extensión de nombre de archivo *html*.

# HTML

- La línea 1 del ejemplo 1 identifica un archivo que contiene una imagen su presentación, mediante su *URL*.
- Línea 2 a la 5 son directivas para comenzar y finalizar un párrafo, respectivamente.
- Las líneas 3 y 4 contienen texto que se mostrará en la página web en el formato de párrafo estándar.
- La línea 4 especifica un enlace en la página web. Contiene la palabra *Moon* rodeada por dos etiquetas HTML relacionada, `< A HREF... >` y `< /A >`.
- El texto entre estas etiquetas es lo que aparece en el enlace tal como se presenta en la página web.

El navegador registra la asociación entre el texto que se muestra en el enlace y el URL contenido en la etiqueta `< A HREF... >` - en este caso la página: Moon

# Uniform Resource Locator: URL

Los navegadores examinan las *URL* para acceder a las correspondientes recursos.

A veces, el usuario escribe una *URL* en el navegador, o el navegador busca la *URL* correspondiente cuando el usuario hace *clic* en un enlace o selecciona uno de sus marcadores o *bookmarks*.

## HTML

Conozca más de URL visitando este sitio: [URL](#)

# URL

- Cada *URL*, en su forma completa y absoluta, tiene dos componentes de nivel superior:

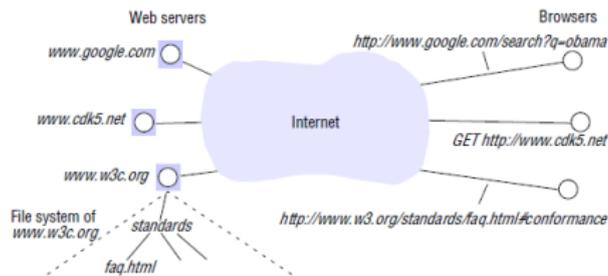
scheme : scheme-specific-identifier

- El primer componente declara qué tipo de *URL* es.
- Los *URL* son necesarios para identificar recursos. Por ejemplo, *mailto: joe@anISP.net* identifica la dirección de correo electrónico de un usuario.
- Las *URL HTTP* son las más utilizadas para acceder a los recursos utilizando el protocolo estándar HTTP.
- Las *URL HTTP* tiene dos funciones principales: identificar qué servidor web mantiene el recurso e identificar cuál de los recursos de ese servidor es necesario.

# URL.

La Figura muestra tres navegadores que emiten solicitudes de recursos a tres servidores web.

- El navegador superior está emitiendo una consulta a un motor de búsqueda.
- El navegador del medio requiere la página predeterminada de otro sitio web.
- El navegador inferior requiere una página web que se especifica en su totalidad, incluido un nombre de ruta relativo al servidor.



# URL

Las *URL HTTP* tienen el siguiente formato, ver figura 3 (de [Coulouris, 2011])

- Una *URL HTTP* completa siempre comienza con cadena *http: //* seguida de un nombre de servidor, expresado como un sistema de nombres de dominio (DNS).
- Opcionalmente, el nombre *DNS* del servidor va seguido del número del puerto en el que el servidor escucha las solicitudes, el puerto 80 por defecto, Luego viene un nombre de ruta opcional del recurso del servidor. Si esto está ausente entonces se requiere la página web predeterminada del servidor.
- La *URL HTTP* termina opcionalmente en una consulta.

*http://servername [:port] [/pathName] [?query] [ #fragment]*

Figura: Formato de los URL.

# HTTP

El Protocolo de transferencia de hipertexto, *HTTP* presenta estas características:

**Interacciones de solicitud-respuesta.** *HTTP* es un protocolo solicitud-respuesta. Define un pequeño conjunto de métodos que pueden realizarse: *GET*, recupera datos del recurso y *POST*, proporciona datos al recurso.

**Tipos de contenido.** Los navegadores no son capaces de manejar todo tipo de contenido. Cuando un navegador realiza una solicitud, incluye los tipos de contenido que prefiere. Cuando el servidor responde, lo tiene en cuenta e incluye el tipo de contenido en el mensaje de respuesta. Se puede configurar las acciones que realizará un navegador para un tipo de contenido.

# HTTP

**Un recurso por solicitud.** Los clientes especifican un recurso por cada solicitud *HTTP*. Los navegadores suelen hacer varias solicitudes al mismo tiempo, para reducir la demora general para el usuario.

**Control de acceso simple.** Cualquier usuario con conectividad a un servidor Web puede acceder a sus recursos publicados de forma predeterminada. Los usuarios pueden restringir el acceso al recurso, al configurar el servidor para que emita una alerta cuando se solicita. El usuario externo al recurso puede acceder el recurso, por ejemplo, escribiendo una contraseña.

Amplie su conocimiento visitando este sitio: [What is HTTP](#)

# XML

- *HTML* es limitado en el sentido de que no es extensible a aplicaciones más allá de la exploración de información, tiene un conjunto estático de estructuras, y están vinculados con la forma en que los datos se deben presentar a los usuarios
- El lenguaje de marcado extensible, *XML* ha sido diseñado como una forma de representar datos estandarizados y estructurados, en formatos específicos de la aplicación.
- Los datos expresados en ese lenguaje son portables entre aplicaciones ya que es autodescriptivo.

# XML

- En el protocolo *HTTP*, los datos *XML* pueden ser transmitidos por las operaciones *POST* y *GET*. Por ejemplo, en la tienda de *amazon.com*, las operaciones del servicio web incluyen una para pedir un libro y otra para comprobar el estado actual de un pedido.
- Cualquier operación sobre un recurso puede ser invocado usando uno de los métodos *POST* o *GET*, con contenido estructurado usado para especificar los parámetros, resultados y respuestas de error de la operación.

Conozca más de XML visitando este sitio: [XML](#)

# Preguntas

## Web

- 1 Los recursos en la World Wide Web y otros servicios se nombran por URL. ¿ Qué denotan las iniciales URL ? Dé ejemplos de tres tipos diferentes de recursos web que pueden nombrados por URL.
- 2 Dé un ejemplo de una URL HTTP. Enumere los componentes principales de una URL HTTP, indicando sus límites e ilustrando cada uno de ellos a partir de su ejemplo.

## References



Maarten Van Steen and Andrew Tanenbaum (2017)

Distributed Systems

*Pearson Education, Inc.*



Coulouris, George and Dollimore, Jean and Kindberg, Tim and Blair, Gordon (2011)

Distributed Systems: Concepts and Design

*Addison-Wesley Publishing Company.*



Andrew Tanenbaum and Van Marteen (2007)

Distributed Systems: Principles and Paradigms

*Pearson Prentice Hall.*



Veríssimo, Paulo and Rodrigues, Luís (2012)

Distributed Systems for System Architects

*Springer.*



Dan Sullivan (2015)

NoSQL for Mere Mortals

# Fin