

# Introducción a la Web

Jesús Arias Fisteus

## Aplicaciones Web (OpenCourseWare, 2023)

uc3m

Universidad **Carlos III** de Madrid

Departamento de Ingeniería Telemática



La **World Wide Web** (WWW, o simplemente la Web) es un espacio de información en el cual los artículos de interés, llamados *recursos*, se identifican mediante identificadores globales llamados Identificador de Recursos Uniforme (*Uniform Resource Identifier, URI*), se encuentran entrelazados mediante hipervínculos, y son accesibles mediante Internet.

- ▶ El cliente utiliza normalmente un *navegador* (Google Chrome, Firefox, etc.) para acceder a un recurso (típicamente, una página Web).
- ▶ El recurso tiene un Identificador de Recurso Uniforme (URI) que lo identifica:
  - ▶ Por ejemplo: <http://www.uc3m.es/Inicio>

- ▶ El navegador sigue una serie de pasos para cargar y mostrar un recurso:
  1. Analiza el URI del recurso.
  2. Obtiene una dirección IP para el nombre de dominio mediante una consulta al Sistema de Nombres de Dominio (DNS).
  3. Abre una conexión TCP con el servidor.
  4. Envía una petición HTTP.
  5. Recibe la respuesta HTTP que contiene el recurso (típicamente, un documento HTML).
  6. Analiza el documento HTML.
  7. Envía peticiones HTTP para el resto de recursos necesarios (hojas de estilos, imágenes, código JavaScript, etc.).
  8. Representa el documento HTML.
  9. Cierra la conexión TCP si no se prevé que sea necesaria en un futuro inmediato.

# Paso 1: analiza el URI

Identificador de Recursos Uniforme (URI)

**http://www.uc3m.es/Inicio**

protocolo

nombre de dominio

ruta

## Paso 2: resuelve el nombre de dominio del servidor

- ▶ Usa el Sistema de Nombres de Dominio (DNS) para obtener una dirección IP que se corresponda con el nombre de dominio del servidor.
  - ▶ P.e. para **www.uc3m.es** se obtiene la dirección IP **176.58.10.138**.

## Paso 3: abre una conexión TCP con el servidor

- ▶ Abre una conexión mediante el Protocolo de Control de Transmisión (*Transmission Control Protocol*, TCP) con la dirección IP del servidor, puerto 80 (puerto por defecto de HTTP).
  - ▶ Se usará esta conexión para enviar peticiones HTTP y recibir sus respuestas.
  - ▶ P.e. se abre una conexión TCP con **176.58.10.138** puerto **80**.

## Paso 4: envía una petición TCP

- ▶ El navegador envía una petición HTTP a través de la conexión TCP:

```
GET /Inicio HTTP/1.1
Host: www.uc3m.es
Connection: keep-alive
Cache-Control: max-age=0
User-Agent: Chrome/62.0.3202.89
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Accept-Encoding: gzip, deflate, br
Accept-Language: es-ES,es;q=0.9,en;q=0.8,en-US;q=0.7
```

## Paso 5: recibe la respuesta HTTP

- ▶ El navegador recibe la respuesta HTTP que incluye el recurso pedido:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=E26E82630...; Domain=www.uc3m.es; HttpOnly
Cache-Control: no-store
Last-Modified: Fri, 10 Nov 2017 11:44:28 CET
Content-Type: text/html;charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 10 Nov 2017 10:44:28 GMT

<!DOCTYPE html>
<html lang="es" class="no-js">
<head>
  <title>Inicio | UC3M</title>
  (...)
```

## Paso 6: analiza el documento HTML

- ▶ Un documento HTML es un fichero de texto compuesto por marcas.
- ▶ Incluye:
  - ▶ Una cabecera: metadatos acerca del documento (p.e. su título).
  - ▶ Un cuerpo: contenido a ser representado, incluyendo párrafos, hipervínculos a otros recursos, listas, tablas, imágenes, vídeos, secciones, títulos de secciones, formularios, etc.

# Ejemplo de documento HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" >
    <title>Main characters in Harry Potter</title>
    <link rel="stylesheet" href="example.css" type="text/css" >
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js" ></script>
    <script src="example.js" type="text/javascript" ></script>
  </head>
  <body>
    <h1>Main characters</h1>
    <p>
      These are the <strong>main characters</strong>
      in Harry Potter:
    </p>
    <ul>
      <li id="harry" ><a href="harry.html" >Harry Potter</a></li>
      <li id="ron" ><a href="ron.html" >Ron Weasley</a></li>
      <li id="herm" ><a href="hermione.html" >Hermione Granger</a></li>
    </ul>
    <div>
      
    </div>
    <div>
      <small>Image by <a href="https://pixabay.com/es/users/6763702/" >Maurygraf at Pixabay</a>
      and used under the terms of the Pixabay content license .</small>
    </div>
  </body>
</html>
```

- ▶ El documento HTML puede hacer referencia a otros recursos, en el mismo servidor o en uno distinto, que son necesarios para representarlo correctamente:
  - ▶ Imágenes.
  - ▶ Hojas de estilos.
  - ▶ Código fuente JavaScript.

## Paso 8: representar el documento HTML

- ▶ El documento HTML contiene la información que debe ser representada.
- ▶ Las hojas de estilos CSS contienen instrucciones acerca de cómo representar visualmente esta información.
- ▶ El código JavaScript añade dinamismo a la página Web.

# Ejemplo de documento CSS

```
body {  
    background: rgb(245, 245, 245);  
}  
  
h1, strong {  
    color: navy;  
}  
  
.image {  
    transition: all 0.5s ease;  
    border: 2px solid black;  
}  
  
.image:hover {  
    border-radius: 50%;  
    transform: rotate(360deg);  
    filter: brightness(200%);  
}
```

- ▶ Los programas JavaScript hacen la página HTML interactiva:
  - ▶ **Reaccionando** a las acciones del usuario en la página Web (manejadores de eventos).
  - ▶ **Modificando** dinámicamente la página Web (API DOM).
- ▶ Los navegadores implementan un conjunto de APIs estándar para facilitar el desarrollo de aplicaciones Web complejas en el lado del cliente:
  - ▶ *Websockets*, eventos enviados por el servidor, interacción con la reproducción de audio y vídeo, acceso a cámara Web y micrófono, almacenamiento de datos en el lado del cliente, notificaciones de escritorio, servicios de geolocalización, vídeo conferencia, gráficos acelerados por *hardware*, acceso a sensores, vibración, arrastrar y soltar, etc.

# Ejemplo de programa JavaScript

```
$(function() {  
    var image = $("img");  
    var canvas = image_to_canvas(image);  
    $("#harry").hover(function() {  
        draw_circle(canvas, 76, 80, 40);  
    }, function() {  
        reset_image(canvas, image);  
    });  
});  
  
var image_to_canvas = function(image) {  
    canvas = $("<canvas>").addClass('image');  
    canvas[0].height = image.height();  
    canvas[0].width = image.width();  
    reset_image(canvas, image);  
    image.replaceWith(canvas);  
    return canvas  
}  
  
var draw_circle = function(canvas, cx, cy, r) {  
    var ctx = canvas[0].getContext("2d");  
    ctx.beginPath();  
    ctx.strokeStyle = "red";  
    ctx.lineWidth = 5;  
    ctx.arc(cx, cy, r, 0, 2 * Math.PI, true);  
    ctx.stroke();  
}  
  
var reset_image = function(canvas, image) {  
    canvas[0].getContext("2d").drawImage(image[0], 0, 0);  
}
```

## Paso 9: cierra las conexiones TCP abiertas

- ▶ El navegador mantiene las conexiones TCP abiertas momentáneamente, y las reutiliza si es necesario enviar nuevas peticiones a los mismos servidores.
- ▶ Cuando determina que no ya será necesaria dicha conexión, el navegador la cierra.

- ▶ Un *servidor Web* es un programa que acepta conexiones TCP entrantes, analiza las peticiones HTTP recibidas y genera las respuestas HTTP apropiadas que incluyen los recursos pedidos.
- ▶ Los recursos pueden ser:
  - ▶ Estáticos: son exactamente iguales para cada petición, y el servidor simplemente los carga desde su sistema de ficheros.
  - ▶ Dinámicos: son generados sobre la marcha para cada petición por un programa que se ejecuta normalmente en el servidor (aunque a veces se ejecuta en el lado del cliente).

- ▶ Se ejecuta una aplicación Web en el lado del servidor.
- ▶ Las rutas HTTP se mapean a distintos componentes de la aplicación, ya sean recursos estáticos o programas que generen recursos dinámicos.
- ▶ La aplicación suele almacenar datos en y recuperar datos desde una base de datos.

## Lenguaje

---

## Algunos entornos bien conocidos

---

PHP

Laravel, CodeIgniter, CakePHP, Symfony

Ruby

Rails

Python

Django, Flask

Java

Java Server Faces, Spring MVC, Vaadin

JavaScript

Express, Meteor, Angular, React

Varios lenguajes

ASP.NET

- ▶ Andrew S. Tanenbaum, David J. Wetherall, *Computer Networks, 5th ed.*, Prentice Hall (2010):
  - ▶ Capítulo 7.3 (The World Wide Web).