

# Programación Avanzada



## Arquitectura y Protocolos de WWW

# Temario



- Historia y Servicios WWW. Evolución y tendencias.
- Principios y componentes del WWW. Arquitectura de la WWW.
- Funcionamiento y organización de Internet y la www.
  - Servidores y Clientes, Browsers. Aplicaciones.
  - Protocolo HTTP, Versiones. HTTP en el modelo TCP/IP.
- Uniform Resource Locations ( URL´s), Dominios, Direcciones, puertos.

# Historia



Historia y Servicios WWW.

1980, Tim Berners-Lee

Organización Europea para la Investigación Nuclear (CERN por sus siglas en inglés)

Primeros buscadores  
Mosaic

# Historia

WWW



# Estándares

WWW



W3C®

# A quien responden los desarrolladores ?

WWW



<https://developer.mozilla.org/es/>

# Evolución y Tendencias



HTML Plano

Páginas estáticas

Páginas Dinámicas

CSS (Cascading Style Sheets)

# Evolución y Tendencias



**Semantics**



**CSS3**



**Multimedia**



**Graphics & 3D**



**Device Access**



**Performance**



**Offline & Storage**



**Connectivity**



# Evolución y Tendencias



HTML5



CSS3

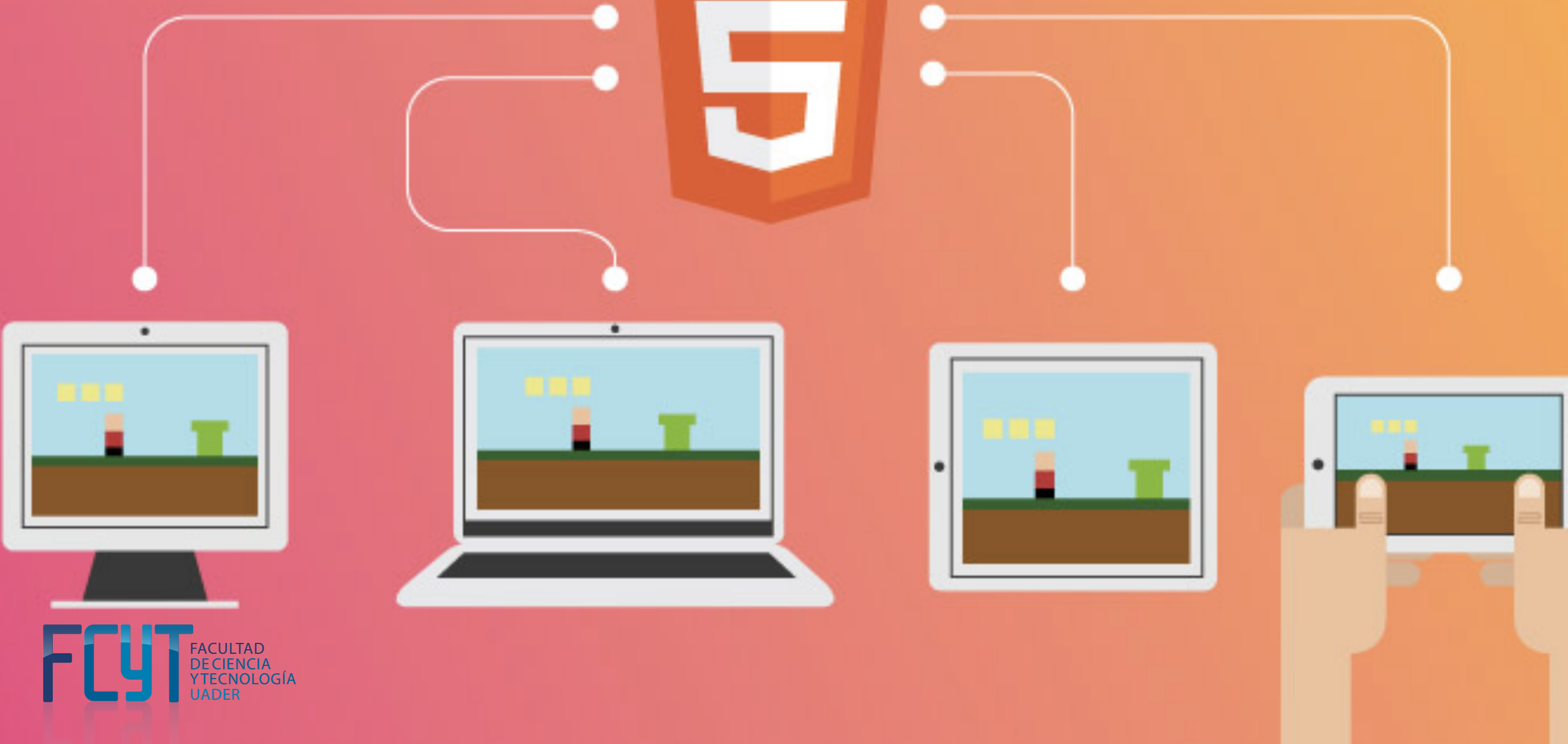


JavaScript

# El poder de Adaptabilidad



**HTML**



# Estructura de una aplicación Web

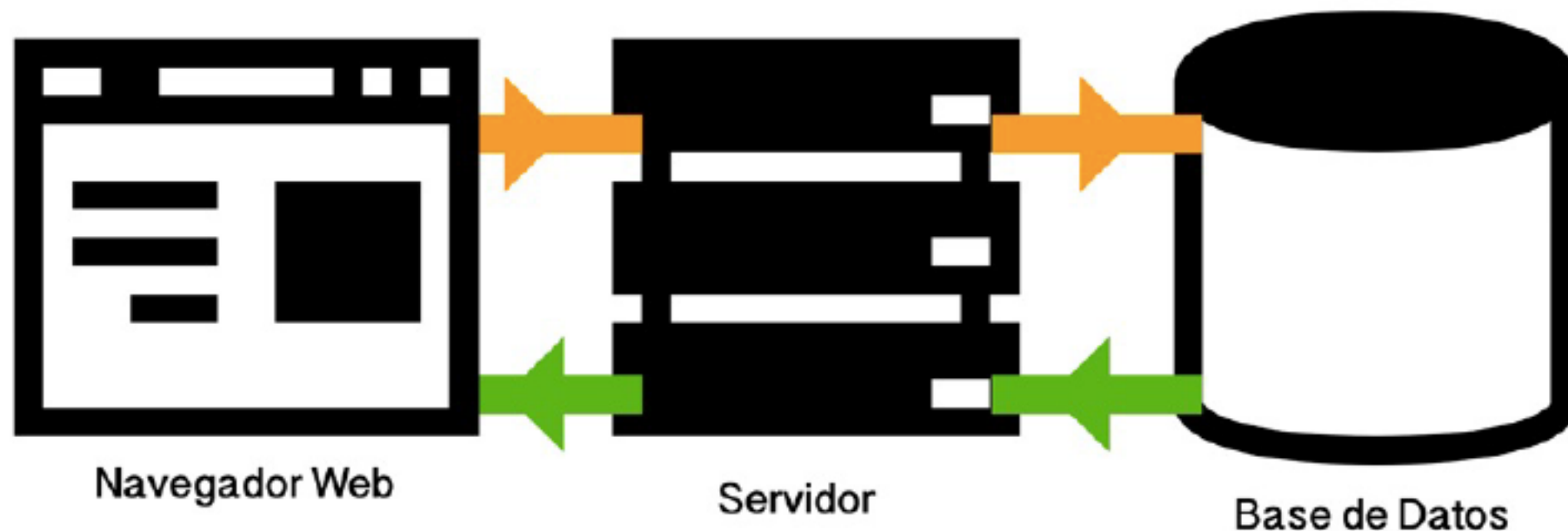


Una aplicación web, actual, está compuesta habitualmente de tres partes principales:

La parte pública o cliente, también conocida como Frontend

La parte del servidor, conocida como Backend

El almacenamiento de datos, o Base de Datos



# Protocolo HTTP



Hypertext Transfer Protocol o HTTP

(en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

# Protocolo HTTP



HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.

# Dialogo HTTP



Para obtener un recurso con el URL `http://www.example.com/index.html`

1. Se abre una conexión al host `www.example.com`, puerto 80 que es el puerto por defecto para HTTP.
2. Se envía un mensaje en el estilo siguiente:

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: nombre-cliente
[Línea en blanco]
```

# Dialogo HTTP



La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página web:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221
```

```
<html>
<body>
<h1>Página principal de tuHost</h1>
(Contenido)
.
.
.
</body>
</html>
```