



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

# Aplicaciones web

**IIC2513 - Tecnologías y Aplicaciones Web**

Sebastián Vicencio R.  
2do Semestre 2020

# Cliente - servidor

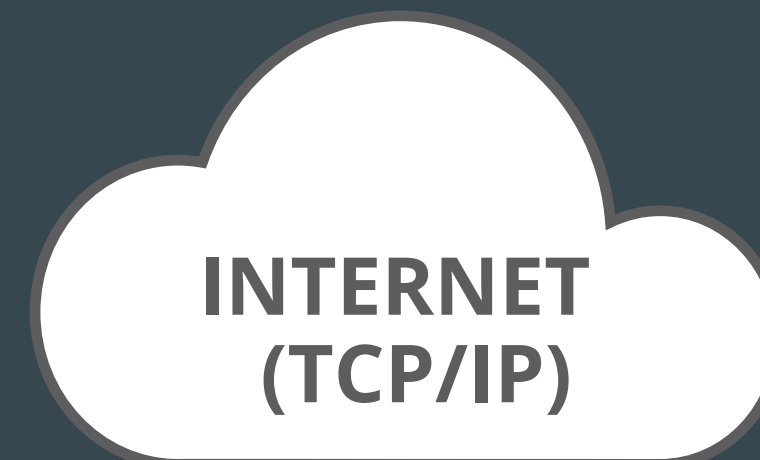
**Evolución de la interacción**

# Evolución cliente - servidor

CLIENTE

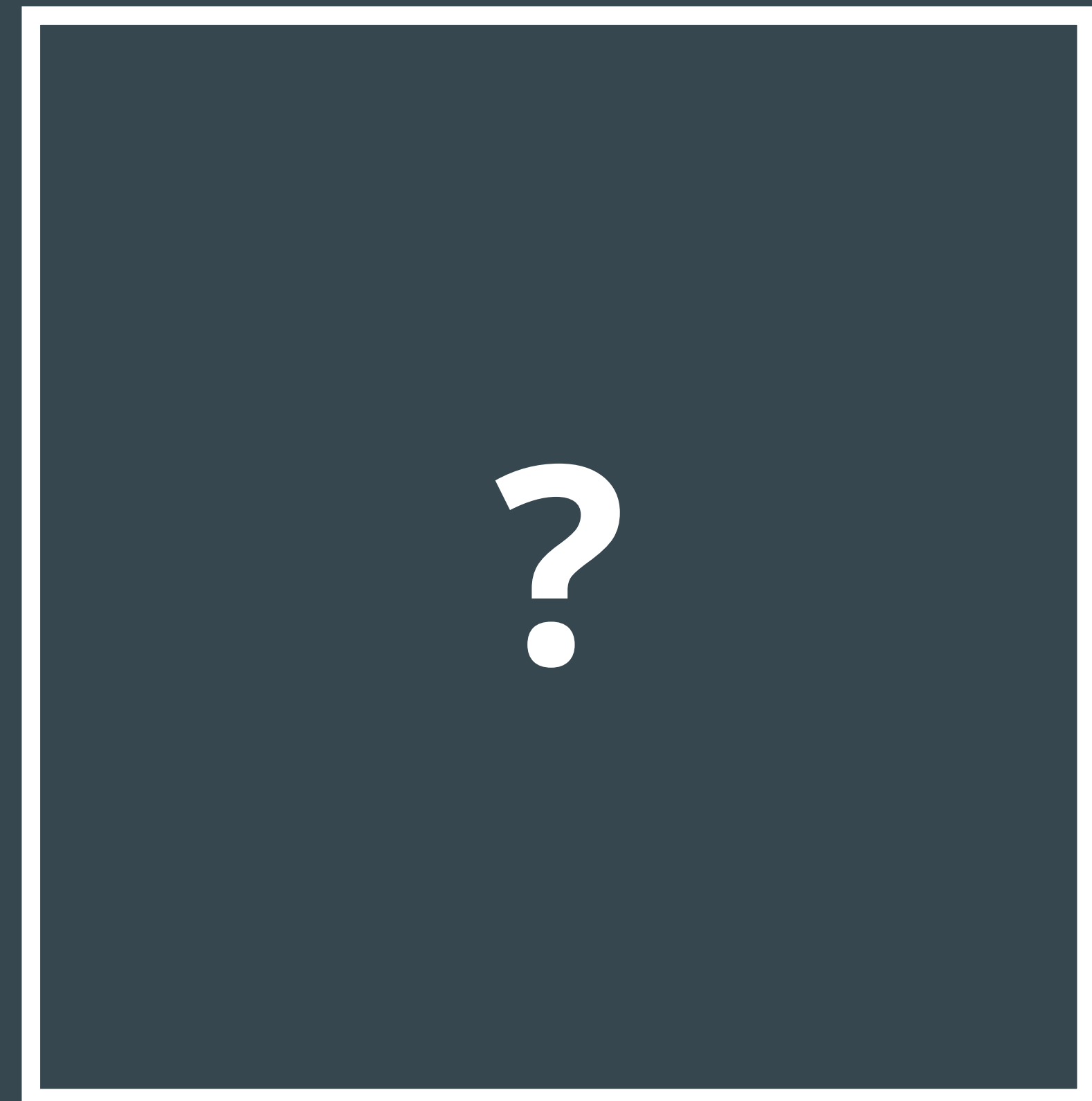


HTTP request



HTTP response

SERVIDOR



# Evolución cliente - servidor

Inicios de la web (1990 - 1993)

CLIENTE

SERVIDOR

<http://example.com/page.html>



HTTP request



HTTP response



Web  
Server

/page.html



# Evolución cliente - servidor

Más documentos que sólo HTML

CLIENTE

SERVIDOR

<http://example.com/img.jpg>



HTTP request



HTTP response



Web  
Server

/img.jpg



# Evolución cliente - servidor

Surge JavaScript en 1995

CLIENTE

SERVIDOR

<http://example.com/script.js>



HTTP request

HTTP response

Web  
Server

/script.js



# Evolución cliente - servidor

Surge CSS entre 1994 - 1996

CLIENTE

SERVIDOR

<http://example.com/style.css>



HTTP request



HTTP response



Web  
Server

/style.css



# Evolución cliente - servidor

Sitio web estático: un documento por página

CLIENTE

SERVIDOR

<http://example.com/page.html>



HTTP request



HTTP response



Web  
Server

/page.html





# Sitio web estático

**Librería con 20 libros**

20 documentos HTML

¿Y si quisiéramos tener **100 libros**?

¿Y si decidiéramos **cambiar la forma de mostrar**  
los libros?

# Evolución cliente - servidor

Ya no sólo documentos, también scripts

CLIENTE

SERVIDOR

<http://example.com/book.php?id=1>



HTTP request

HTTP response

Web  
Server

mod\_php

/book.php?id=1

Genera documento HTML



# Evolución cliente - servidor

Ya no sólo documentos, también scripts

CLIENTE

SERVIDOR

<http://example.com/cgi-bin/script.cgi>



HTTP request

HTTP response

Web  
Server

Run /cgi-bin/script.cgi



Output: documento HTML



mod\_php

# Evolución cliente - servidor

Sitio web dinámico: documento HTML generado

CLIENTE

SERVIDOR

<http://example.com/my-book>



HTTP request



Web  
Server

HTTP response



Run script



Documento HTML



# Evolución cliente - servidor

Complejidad crece (fines 90 - principios 2000)

CLIENTE

SERVIDOR

<http://example.com/my-book>



HTTP request



Web  
Server

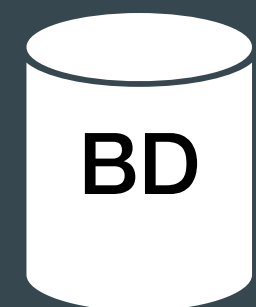
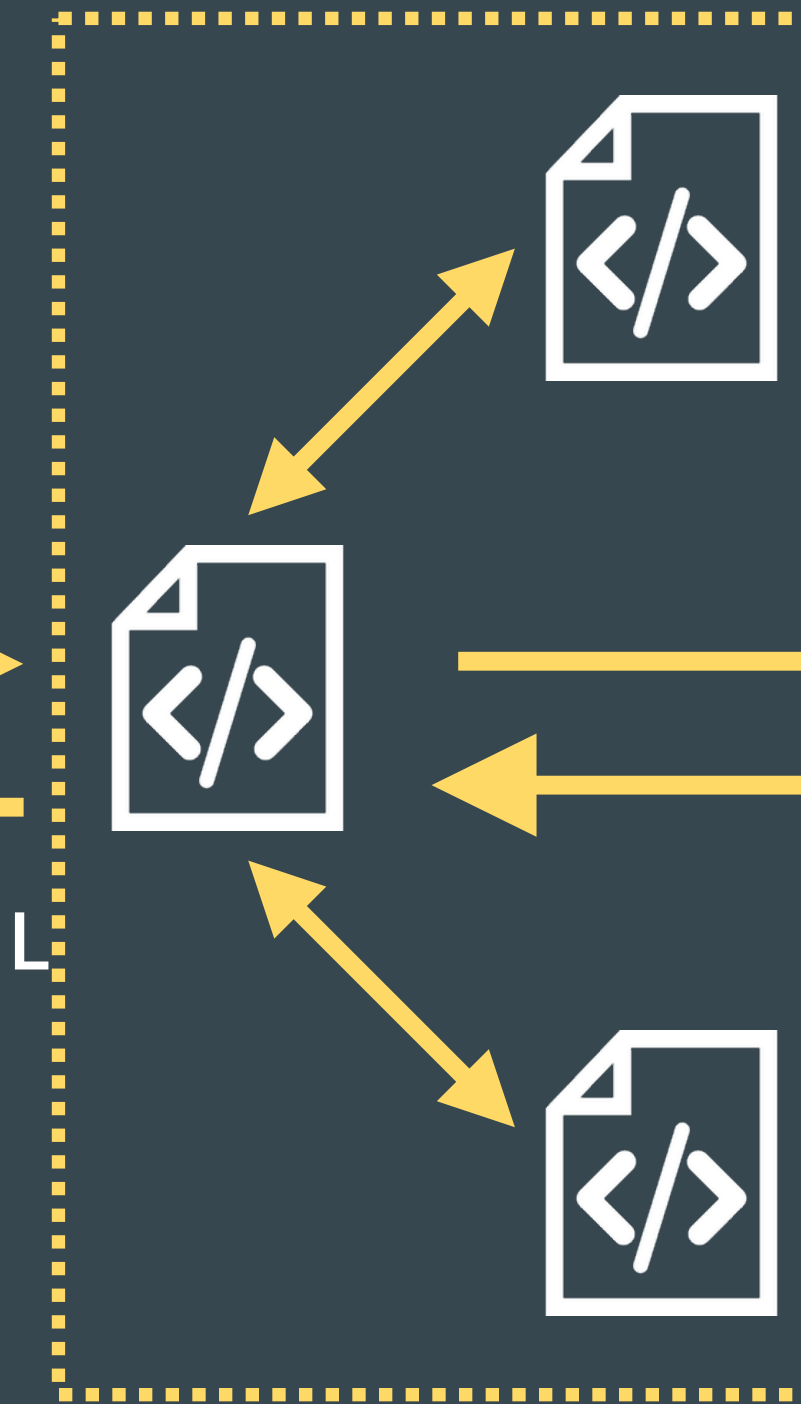
Run script



Documento HTML



HTTP response



# Evolución cliente - servidor

Surge la aplicación web

CLIENTE

SERVIDOR

<http://example.com/my-book>



HTTP request

HTTP response

Web  
Server

Forward

Documento  
HTML

Web  
app

BD

# Aplicación web

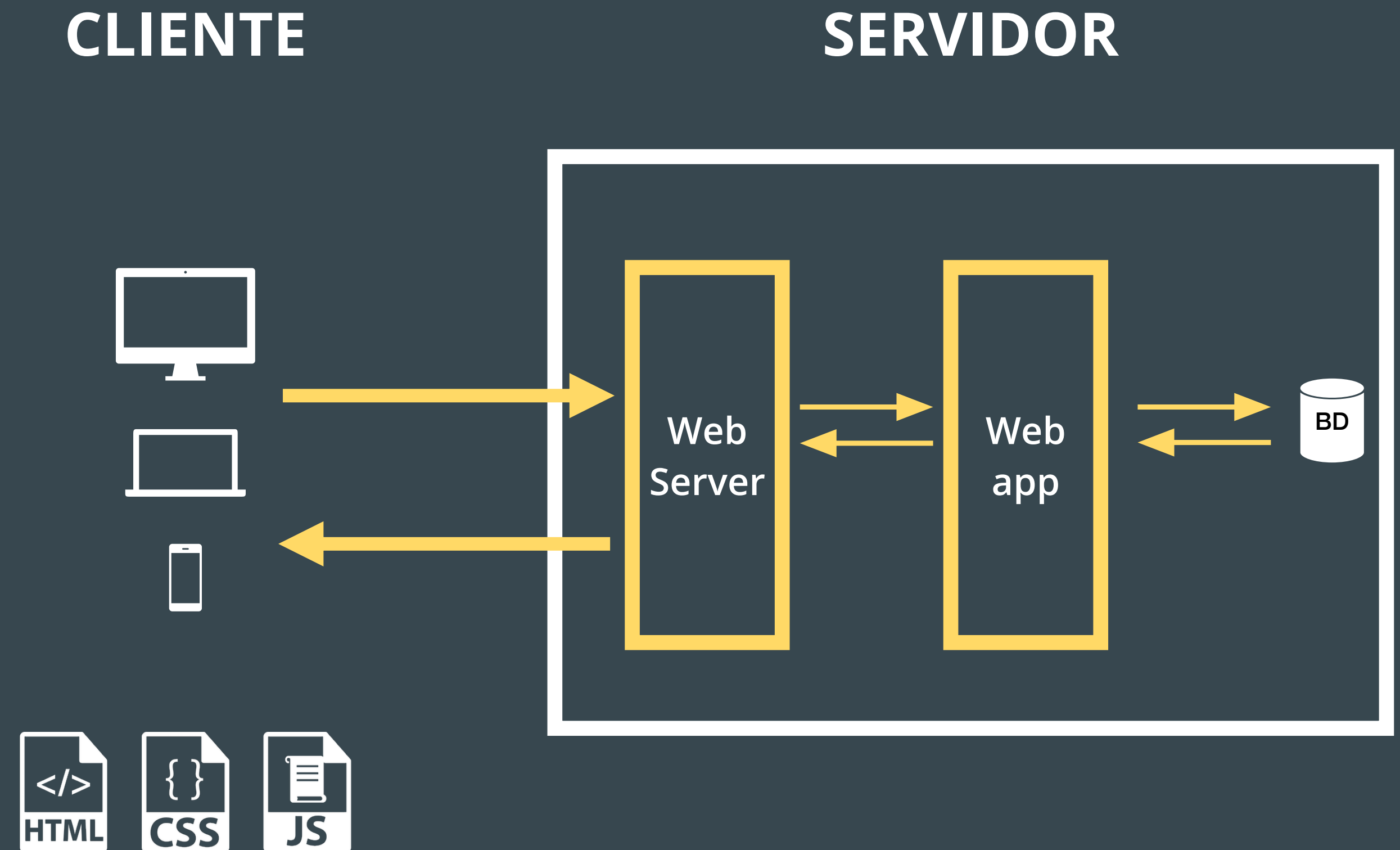
Aplicación que **realiza acciones en el servidor** en base a un request HTTP

**Construye de forma dinámica una respuesta**, la cual se envía de vuelta como un response HTTP

# Aplicación web

## Coordinación entre cliente y servidor

1. Request recurso web
2. Servidor consulta BD
3. Servidor genera documento HTML
4. Cliente recibe e interpreta documento HTML
5. Documento HTML tiene referencia a CSS
6. Request al servidor del documento CSS
7. Servidor devuelve documento CSS
8. Cliente despliega sitio web con estructura y estilo
9. Documento HTML tiene referencia a script JS
10. Request al servidor del script JS
11. Servidor devuelve script JS
12. Cliente recibe y ejecuta script JS





# Aplicación web

En términos de tecnologías

## CLIENTE

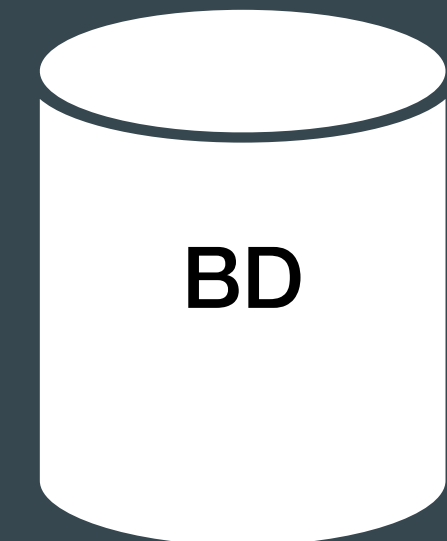


## FRONTEND

Cómo se ve  
Look & Feel

## SERVIDOR

Aplicación web  
(Node.js, Rails, Django)



## BACKEND

Cómo funciona  
Lógica de negocio

# Backend

¿Qué acciones se realizan aquí?

# Backend

¿Qué podemos hacer aquí?

Buscar y/o persistir  
información en una BD

Manejar sesiones de  
usuario

Controlar acceso a  
contenido

Conexión con servicios  
(correo, transacciones)

Y lo que se nos ocurra...

# Proyecto curso

**Aplicación web**

# Proyecto curso

## Roadmap

1. **Backend:** funcionalidad básica
2. **Frontend:** páginas bien construidas y estiladas
3. **Backend:** manejo de usuarios y protección contenido
4. **Backend y Frontend:** funcionalidad - UI terminadas
5. **Frontend:** comportamiento con JavaScript
6. **Backend y/o Frontend:** REST API

# Proyecto curso

Primera entrega: diseño de la solución

**Actores y casos de uso**

**Mockups y navegación**

**Modelo de datos**

# Actividad

**Hands-on: mi primer sitio web**