

# **Aplicaciones Distribuídas en Internet**

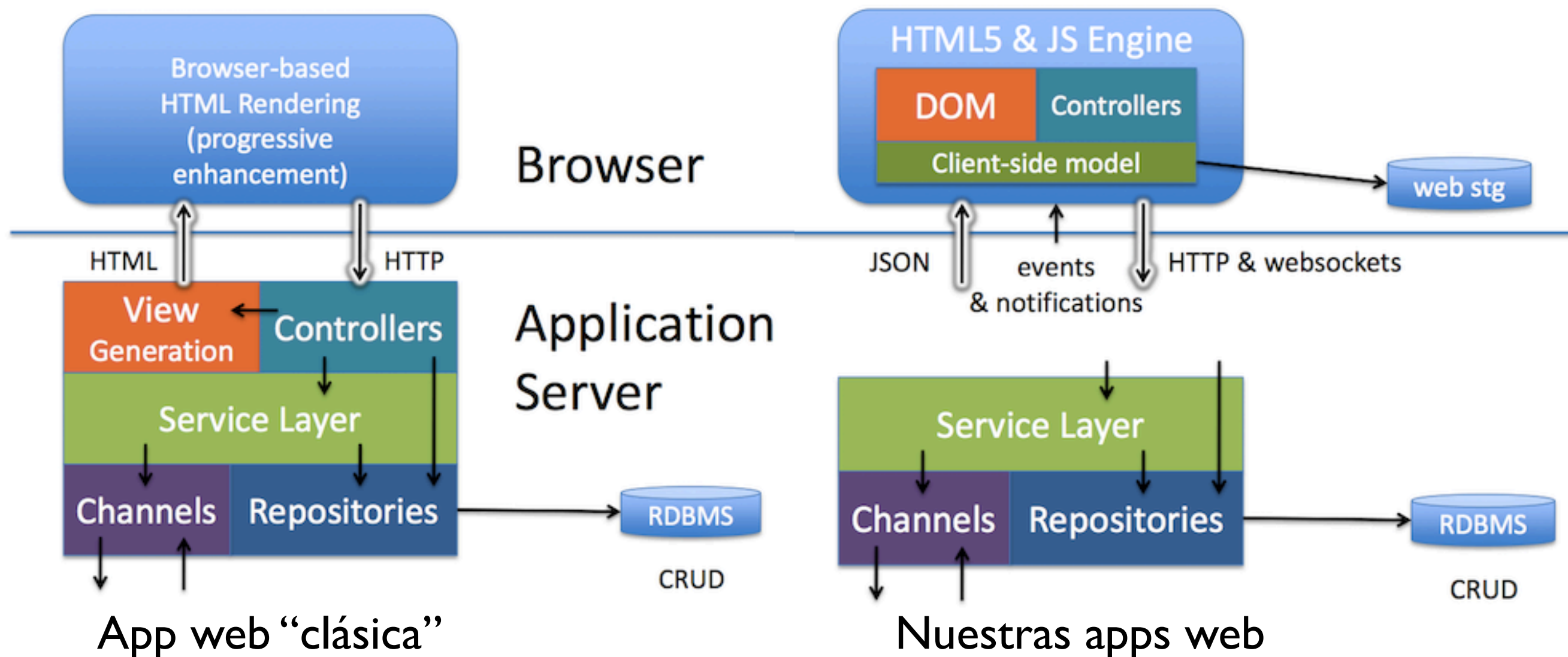
Presentación y normas básicas 2020/21  
Otto Colomina Pardo - [otto@dccia.ua.es](mailto:otto@dccia.ua.es)

**Contenidos**  
**Organización**  
**Evaluación**

# Contenidos

# La asignatura

Desarrollo de aplicaciones basadas en **servicios web REST**. El servidor ofrece un **API remoto** al que accede la aplicación cliente en el navegador, que será una **Single Page Application**



# Bloques del temario

## 1. Servicios web REST. Aplicaciones en el servidor (4 semanas)

- HTTP y aplicaciones web. Introducción a REST
- Diseño de APIs REST
- Autenticación en REST

Backend

## 2. Aplicaciones web en el cliente (7 semanas)

- Javascript en clientes web. Eventos . APIs del navegador
- Frameworks JS
- Gestión del estado de la app
- Nuevos paradigmas: functional reactive programming

Frontend escritorio

## 3. Aplicaciones web en dispositivos móviles (4 semanas)

- APIs JS en dispositivos móviles
- PWAs

Frontend móviles

# Prácticas y duración aproximada

- Prácticas **evaluables**

1. Diseño e implementación de un API REST (aprox. 3 semanas)
2. Aplicaciones web en el cliente con Javascript estándar (1 semana)
3. Aplicaciones web en el cliente con Frameworks JS (aprox. 3 semanas)
4. Aplicaciones web para dispositivos móviles (aprox. 2 semanas)

- Sesiones **guiadas**

- Introducción a Javascript
- APIs REST de tipo CRUD con Node.js y Express
- Introducción a frameworks JS de escritorio (Vue)
- Introducción a frameworks JS para móviles

# Asignaturas relacionadas

## Cursos anteriores

- Herramientas Avanzadas para el Desarrollo de Aplicaciones

*Desarrollo de aplicaciones web en .NET*

## Itinerario

- Ingeniería Web

*Lenguaje de modelado web. Desarrollo de aplicaciones web*

- Metodologías y Tecnologías de Integración de Sistemas

*Servicios web SOAP y REST*

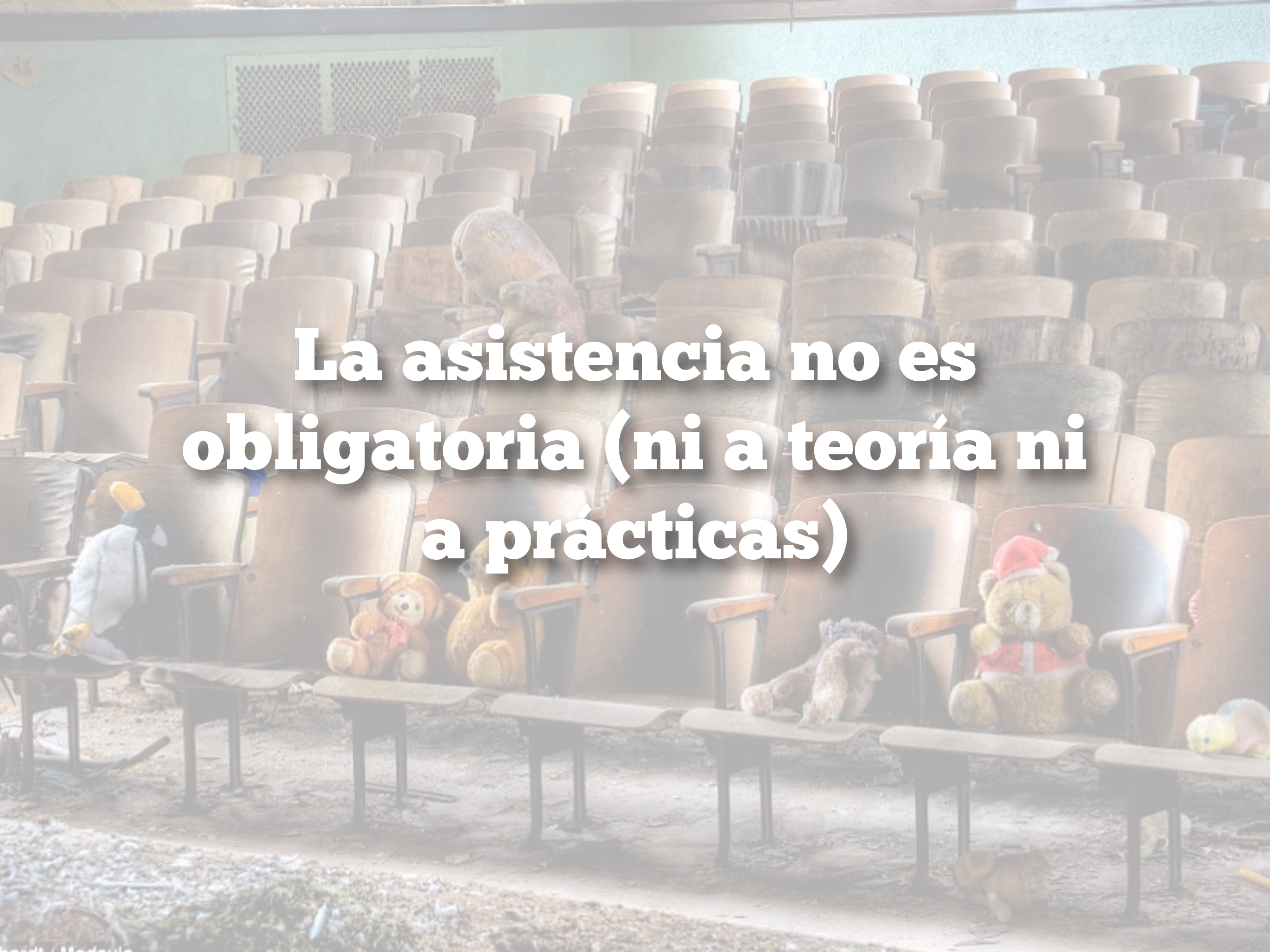
# Organización



# Docencia dual 🙄: tiempo de clase

- **Explicaciones teóricas** (aprox. 40 minutos)
  - Se grabarán y dejarán en la web si es posible
  - Para complementar, algunas veces tendréis que ver algún video corto o leer algún tutorial por vuestra cuenta, se intentará dejar tiempo de clase
- **Ejercicios** (aprox. 30 minutos)
  - Ejercicios a realizar de manera individual en clase
- **Preparación de los trabajos** en grupo (aprox. 30 minutos)
  - Búsqueda/estudio de material
  - Resolver dudas
  - Puesta en común (dentro de lo posible... 🙄)





**La asistencia no es  
obligatoria (ni a teoría ni  
a prácticas)**

# Evaluación

# Evaluación

- Prácticas (individuales) (60%)
- Ejercicios de clase (individuales) (20%)
- Trabajos (por grupos) (20%)



# Prácticas (60%)

*(La ponderación es aproximadamente proporcional al tiempo dedicado en la asignatura)*

- Desarrollo de un API REST 35%
- JS estándar 10%
- Frameworks JS en el escritorio 35%
- Aplicaciones web para móviles 20%

# Ejercicios en clase (20%)

- El objetivo es consolidar los conceptos vistos en clase de teoría
- Generalmente serán de programación (**con ordenador**) aunque algunos pueden ser sobre papel
- Se deben realizar y terminar **durante el horario de clase** (se entregan en la web), **no se admitirán entregas posteriores**

# Trabajos en grupo (20%)

- Trabajos sobre temas complementarios o que no se pueden ver en profundidad en la asignatura
- 2 trabajos durante el curso, a desarrollar en **grupos de 2-3 personas**. Reservaremos un tiempo de cada clase a avanzar en el trabajo
- Idealmente un trabajo debería tener una parte de conceptos y otra práctica tipo tutorial/demo, aunque podría haber trabajos totalmente teóricos
- Entregable del trabajo: video (Youtube o similar) y/o documento
- Estas primeras semanas no empezamos todavía, más adelante se publicará
  - Una lista de posibles temas, aunque podéis proponer los vuestros
  - Una guía de evaluación (p. ej. Se valorarán más los trabajos con parte práctica, con video, ...)