

# **Aplicaciones Distribuídas en Internet**

Presentación y normas básicas 2021/22  
Otto Colomina Pardo - [otto@dccia.ua.es](mailto:otto@dccia.ua.es)

**1. Contenidos**

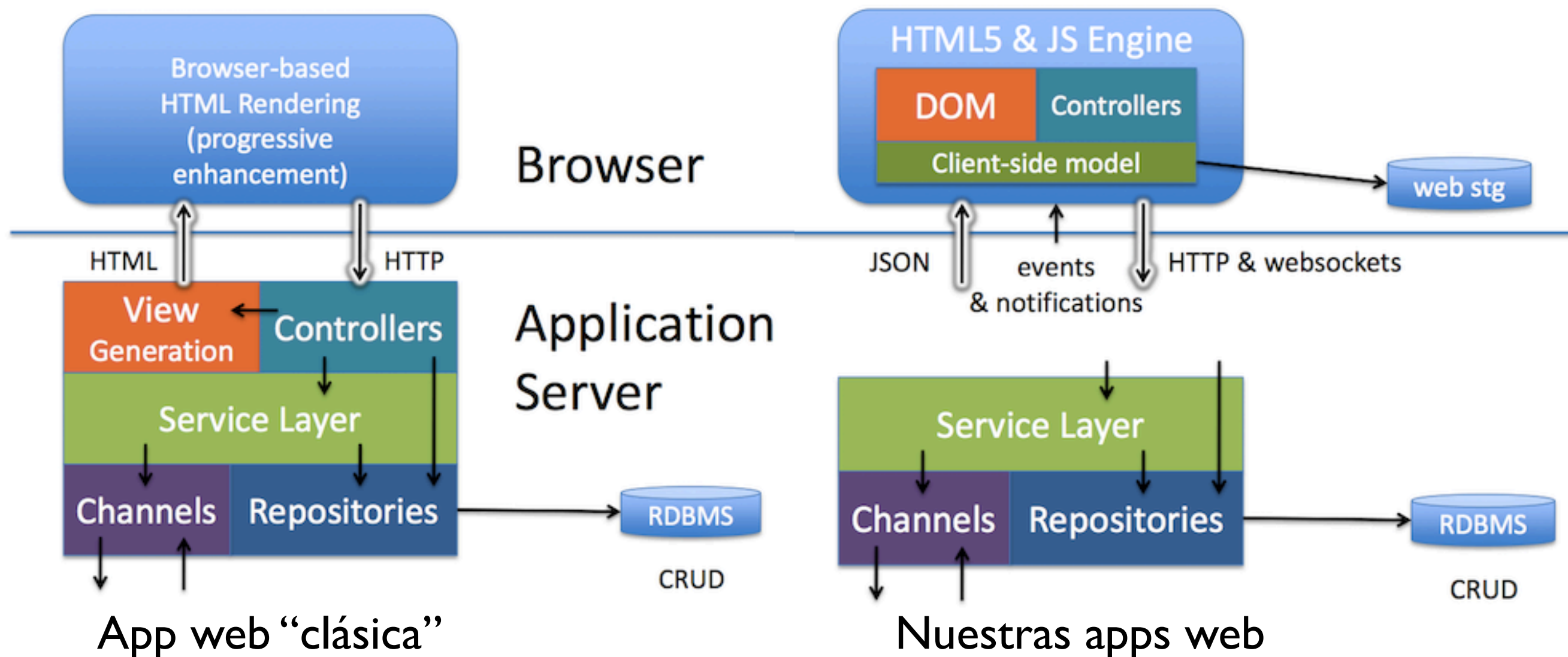
**2. Organización**

**3. Evaluación**

# 1. 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. APIs web (aprox. 4 semanas)

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

Backend

## 2. Aplicaciones web en el cliente (aprox 6 semanas)

- Javascript en clientes web. Eventos . API DOM
- Comunicación con el servidor
- Arquitecturas de apps web en el cliente
- Frameworks JS en el cliente

Frontend escritorio

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

- Introducción. APIs JS en dispositivos móviles
- PWAs

Frontend móviles

## 4. Introducción a las arquitecturas en el lado del servidor (1 semana)

Backend

# 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*

- Sistemas distribuidos

*APIs REST*

## Itinerario

- Ingeniería Web

*Lenguaje de modelado web. Desarrollo de aplicaciones web*

- Metodologías y Tecnologías de Integración de Sistemas

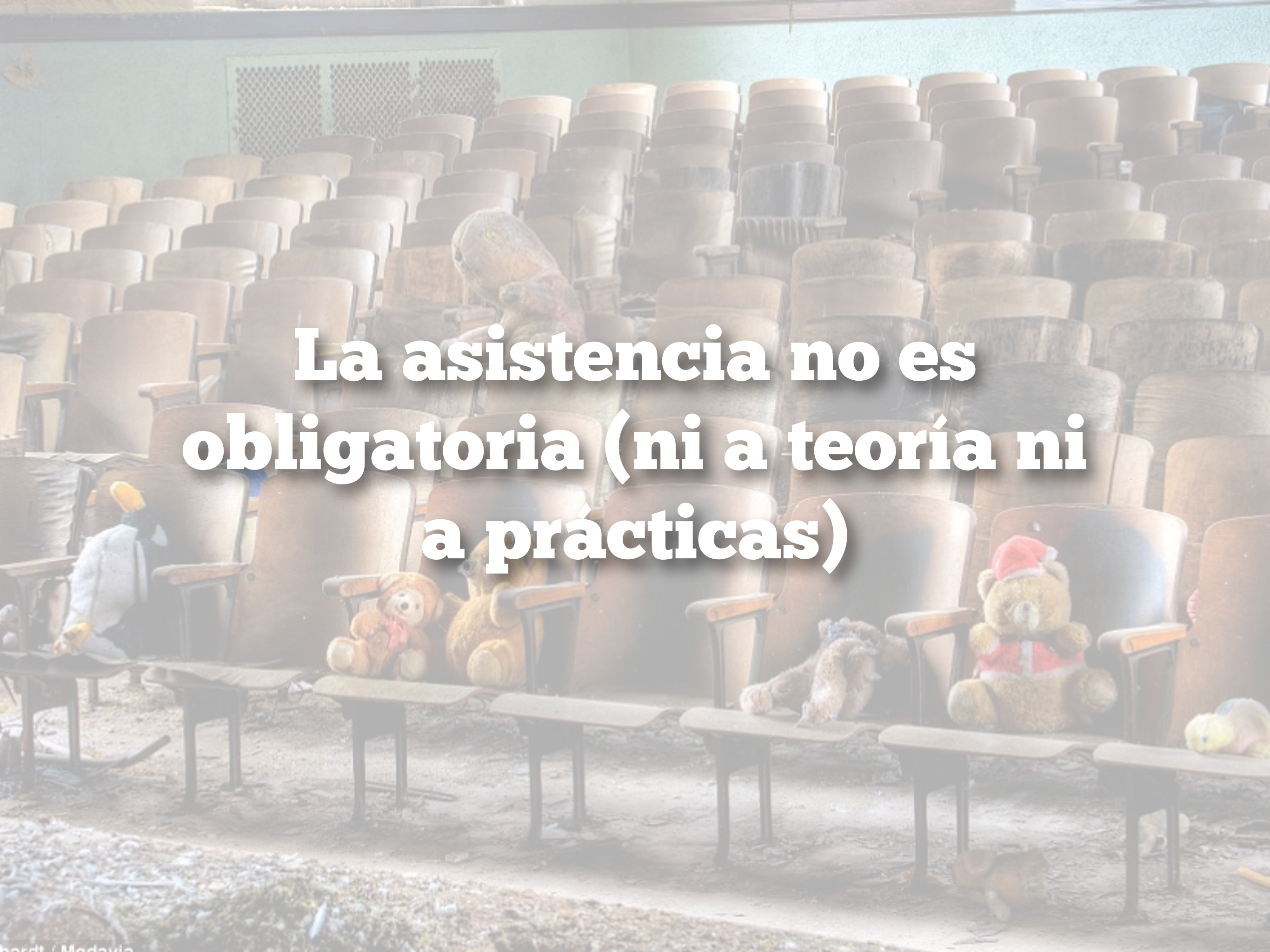
*Servicios web SOAP y REST*

## **2. Organización**



# Tiempo de clase

- **Explicaciones teóricas** (aprox. 60 minutos)
- **Ejercicios** (aprox. 40 minutos)
  - Ejercicios a realizar de manera individual en clase, se entregan al final de la misma
- Algunas clases (pocas) se dedicará un tiempo de **discusión de los trabajos** en grupo
  - Resolver dudas
  - Puesta en común entre el grupo y con el profesor



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



# 3. Evaluación

# Evaluación

- Prácticas (65%)
- Ejercicios de clase (20%)
- Trabajos grupales (15%)

# Prácticas (65%)

*(La ponderación de cada práctica 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**
  - Si alguien no puede asistir a clase **por un motivo justificado** que se ponga en contacto conmigo y veremos cómo es posible recuperar este 20% de la nota

# Trabajos en grupo (15%)

- Trabajos sobre temas complementarios o que no se pueden ver en profundidad en la asignatura
- 1 trabajo durante el curso, a desarrollar en **grupos de 2-4 personas**.
- 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** y/o **documento**
- Se publicará en el moodle
  - Una lista de posibles temas, aunque podéis proponer los vuestros
  - Un baremo de evaluación (p. ej. Se valorarán más los trabajos con parte práctica, con video, ...)