

# Prácticas de Construcción de Servicios y Aplicaciones Audiovisuales en Internet

Grado Ing. Sistemas Audiovisuales y Multimedia  
Universidad Rey Juan Carlos, Curso 2016-2017  
versión 0.2

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Práctica Html5 básico</b>	<b>2</b>
2.1. Ingredientes obligatorios . . . . .	2
<b>3. Práctica JavaScript: Puzzle</b>	<b>2</b>
<b>4. Práctica JavaScript: Juego del Comecocos</b>	<b>4</b>
4.1. Ingredientes obligatorios . . . . .	6
<b>5. Práctica Video y Audio en cliente: Realizador de TV</b>	<b>6</b>
5.1. Funcionalidades obligatorias . . . . .	7
<b>6. Práctica Video y Canvas: filtro de color de un video</b>	<b>7</b>

## Introducción

Este documento presenta el enunciado de las prácticas de la asignatura Construcción de Servicios y Aplicaciones Audiovisuales en Internet. Las prácticas no se entregan, aunque es muy recomendable realizarlas todas. Son preparatorias para el *examen de prácticas*, que es el que os evaluamos. En él os pediremos que hagais alguna aplicación web, con alguna variante respecto a lo que habeis hecho en estas prácticas preparatorias.

Las prácticas cubren muchos aspectos del temario de teoría

IMPORTANTE: las prácticas puedes prepararlas en tu ordenador, pero han de funcionar *obligatoriamente* en el laboratorio de la ETSIT.

## Práctica Html5 básico

Crea tu Curriculum Vitae como una página web estática. Utiliza elementos típicos de html5: tablas, una imagen, etc.. Agrupa varios elementos que tienen sentido juntos en `<div>` y asócialos un estilo de visualización propio (clase). Maneja hojas de estilos de CSS3, definidos en la sección `<head>` de la página para mejorar su apariencia y aplica algún efecto de CSS3.

### Ingredientes obligatorios

- Etiquetas semánticas y de estructura de documento que están disponibles en html5 pero no lo estaban en html4.01.
- Estilos CSS3
- Efectos CSS3
- `<div>` y `<span>`

## Práctica JavaScript: Puzzle

El objetivo de esta prueba es que practiques con JavaScript, centrándote en el lado cliente de una aplicación web. Aunque los contenidos vienen desde un servidor web, típicamente estas aplicaciones hacen hincapié en código que se ejecuta en el navegador, con muchas interacciones locales con el usuario humano.

Programa un rompecabezas visual, donde varias imágenes se han dividido en piezas regulares que se pueden mover al picar con el ratón sobre ellas. Se ha dejado una pieza vacía y sólo se pueden mover las vecinas a ese hueco. Las piezas aparecen desordenadas. Tendrás que modificar el DOM de la página desde tu código Javascript según las iteraciones del usuario humano cuando pica en una zona u otra del puzzle. Si pica en una zona que no es vecina del hueco vacío nada debe cambiar.



Figura 1: Ejemplo de CV generado con Html5

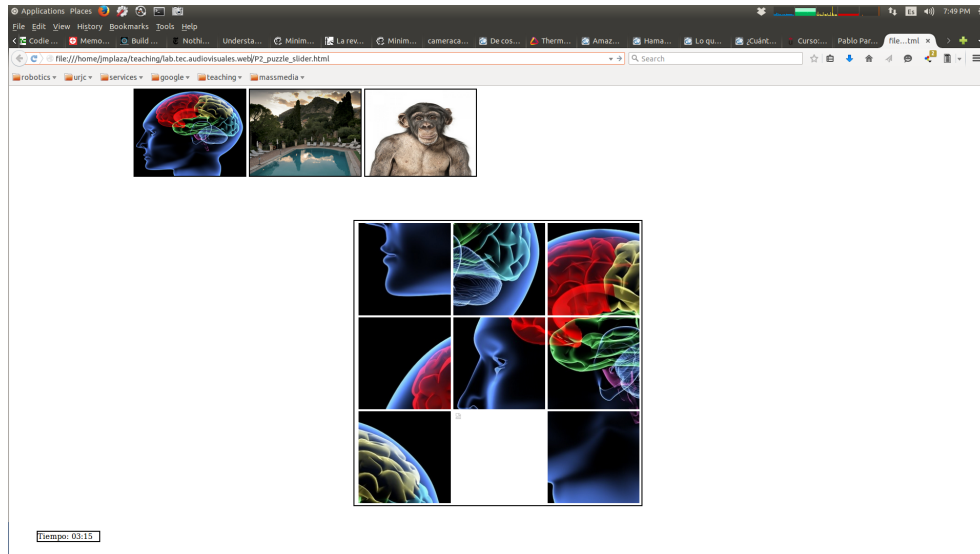


Figura 2: Aplicación web del puzzle

La aplicación medirá el tiempo que el usuario tarda en completar el puzzle desde que se le enseña. Utiliza para ello algún temporizador propio de Javascript (SetInterval).

La aplicación mostrará también un carrousel de imágenes en pequeño tamaño de modo que el usuario puede elegir de qué imagen hará el rompecabezas. Las imágenes del carrousel van rotando periódicamente y muestra 3 de las 10 en el catálogo de puzzles posibles.

Se incluirá un manejo de cookies sencillo, usándolas desde el lado cliente para almacenar usuarios reconocidos y su tiempo. Si un usuario bate su propio record resolviendo el rompecabezas en menor tiempo, la página web le mostrará un cuadro de diálogo felicitándole.

## Práctica JavaScript: Juego del Comecocos

El objetivo de esta prueba es que practiques con JavaScript, manejo de DOM y de BOM. Aunque los contenidos de la página web vienen desde un servidor web, típicamente aplicaciones como ésta hacen hincapié en el código que se ejecuta en el navegador, con muchas interacciones locales con el usuario humano, es decir, en el lado cliente de una aplicación web.

Programa el juego del **comecocos** como aplicación web. Se ofrecerá un laberinto con paredes (no es necesario que sean redondeadas las paredes). En el escenario habrá un comecocos que debe ser teleoperable por el humano con el teclado, que lo va guiando pulsando algunas teclas. No puede moverse en diagonal.

Tienes un tutorial interesante sobre cómo manejar elementos de html5 para videojuegos en <http://www.w3schools.com/games/default.asp>

### Comecocos elegible

En una parte de la página, fuera del escenario habrá tres comecocos, de tres colores distintos y un recuadro. El usuario humano puede elegir con qué comecocos concreto jugar arrastrando alguno de esos

tres comecocos posibles hasta el recuadro. Empieza por defecto con un comecocos amarillo. Cuando el usuario termina de seleccionar uno se restauran las tres opciones posibles. Utiliza el API **drag & drop** para materializarlo.

## Fantasma

Habr  dos **fantasmas**, que se mover n aut nomamente de modo continuo en l nea recta. Se mover n en los dos grandes pasillos del circuito cambiando hacia la direcci n opuesta cuando choquen con una pared. Su comportamiento no var a cuando el comecocos come un punto gordo, siguen exactamente igual.

## Frutas

Dentro del escenario habr  frutas que el comecocos puede comer. Saldr n como mucho cuatro, siempre en la misma posici n del laberinto. Saldr n en un intervalo aleatorio entre (0 y 10 segundos) una vez que la anterior fruta ha sido comida.



Figura 3: Aplicaci n web del comecocos

## Tiempo

El juego tiene una duraci n m xima, un Tiempo M ximo Permitido. Habr  un bot n de inicio y otro de pausa+contin a (es un  nico bot n). Cuando se pulsa la pausa el juego y el tiempo para la puntuaci n se detienen. La aplicaci n medir  el tiempo que el comecocos tarda en completar el juego desde que el usuario pulsa el bot n de comienzo. Utiliza para ello alg n temporizador propio de JavaScript (**SetInterval**).

## Puntuación y records

A medida que el comecocos come los puntos o frutas va acumulando puntos. Cada punto pequeño suma 1 a la puntuación, y cada punto gordo 3. Al final del juego el número de segundos restantes hasta el Tiempo Máximo permitido se suma a los puntos, y esa es la puntuación final.

La puntuación temporal se muestra en pantalla. La calcula un **WebWorker**, al que se le notifican sucesos (pacman se come un punto normal, se come un punto gordo, se come una fruta, ha pasado un segundo...) y entrega la puntuación al hilo principal en un mensaje.

La aplicación web guardará y mostrará en todo momento, fuera del escenario, dos records: el de la sesión y el absoluto de siempre. Utiliza las APIS de **SessionStorage** y de **LocalStorage** para materializar esto.

## Sonido y video

Durante el juego habrá un sonido de continuo y cuando pasa encima de una fruta, se oirá brevemente un sonido especial. Utiliza el elemento `< audio >` para ello.

Cuando se termine el juego, bien porque el comecocos se come todos los puntos, bien porque le atrapa un fantasma, o bien porque expira el tiempo máximo, se reproducirá un video en la página, fuera del escenario. En el primer caso un video con aplausos y en los otros dos, un video de desilusión.

## Ingredientes obligatorios

- Manipulación del DOM
- Canvas 2D
- Motor temporal, `setInterval`
- Teleoperación desde teclas
- WebWorkers
- LocalStorage y SessionStorage
- API Drag and Drop
- Audio y video.

## Práctica Video y Audio en cliente: Realizador de TV

Programa una aplicación web para un realizador de televisión, que tiene varios videos disponibles y debe seleccionar cuál de ellos mostrar en el video final de salida en cada momento. Típicamente estas cámaras observan el mismo espectáculo en directo desde diferentes perspectivas. Dependiendo del pasaje del espectáculo el realizador puede elegir qué cámara prefiere en el montaje final y cambiarlo a medida que el espectáculo se desarrolla. Este video de salida es el que la cadena de televisión realmente difundiría en antena.



Figura 4: Realizador de televisión, contexto

Esta práctica pone el foco en el manejo de los elementos video y audio desde el lado cliente de la aplicación web, manipulando el DOM asociado a estos elementos en html5.

### Funcionalidades obligatorias

- cambio entre distintos videos (camara1,camara2,camara3,camara4) y en un video nuevo mostrar la fuente elegida, cambiabile a voluntad desde el GUI. Se debe recuadrar la fuente activa con un color diferente, de modo que sea intuitivo saber de un vistazo cual de ellas se está emitiendo.
- debe mostrar un reloj digital con el tiempo de reproducción, en grande.
- marcar puntos en el video para hacer bucles. Por ejemplo: si los videos duran 5 minutos y se quiere un bucle entre el minuto 3 y 4, cuando la reproducción llegue al 4 debe retroceder al minuto 3 y continuar. Debe poder configurarse desde la página web cuantas vueltas ha de dar ese bucle antes de que la reproducción salga de él.
- que el sonido en un video sólo se oiga cuando el ratón en la página web está justo encima del video. Algo parecido utilizan los anunciantes en las páginas web de los periódicos.

### Práctica Video y Canvas: filtro de color de un video