



Programación Avanzada
Grado en Ingeniería Informática en Sistemas de Información - Curso 2017/2018
EPD 10: Extensiones de jQuery

La entrega del trabajo se hará a través de la tarea correspondiente en el Campus Virtual. Pasado el límite de entrega se aceptará el envío del trabajo, con una penalización de 2 puntos sobre 10 de la calificación por cada hora o fracción de retraso. La entrega consistirá en un único fichero comprimido en formato ZIP cuyo nombre deberá ser de la forma *equipoXX.zip*, donde *XX* serán dos cifras que indicará el número del equipo. Por ejemplo, *equipo07.zip*. Este fichero contendrá una serie de carpetas cuyo nombre deberá ser de la forma *ejY* o *pZ*, donde *Y* y *Z* representan, respectivamente, el número de cada ejercicio o problema del presente guión. Dentro de dichas carpetas se incluirán exclusivamente los archivos necesarios en la resolución del correspondiente ejercicio o problema. Las rutas de los ficheros empleados serán relativas, a fin de que las resoluciones a los ejercicios y problemas puedan ser examinadas en cualquier equipo. Cualquier entrega que no cumpla las reglas de nombrado, el formato de compresión del archivo o el contenido de los archivos del mismo, será penalizada con 2 puntos sobre 10 por cada incumplimiento.

Objetivos

- Entender el concepto de extensión o *plugin* de jQuery.
- Hacer uso de los *plugins* de jQuery para añadir comportamientos dinámicos a páginas web.
- Desarrollar *plugins* básicos.

Conceptos

Una de las prácticas más frecuentes dentro de la programación es reutilizar el código ya creado, tanto por nosotros mismos como por otros desarrolladores (restringido siempre por las diferentes licencias de los códigos de terceros). Dentro de este proceso de reutilización también debe contemplarse la posibilidad de adaptar, completar y mejorar esos componentes a la realidad y necesidades de nuestras aplicaciones.

En esta línea, el *framework* jQuery incorpora las llamadas extensiones o *plugins*. En jQuery, un *plugin* no es más que un método creado para extender las funcionalidades que proporciona la librería central de jQuery, siempre con la idea de realizar una determinada acción sobre una colección de elementos. De esta manera, al acceder a un objeto jQuery (\$) éste hereda todos los métodos del objeto \$.fn incluidos en la librería jQuery más todos los métodos de los *plugins* que hayamos incorporado, aumentando de esta forma el potencial de la librería jQuery.

Como desarrolladores, tenemos la posibilidad de usar los *plugins* de jQuery disponibles o incluso crear nuevos *plugins* para usarlos de forma privada o publicarlos para que estén a disposición de otros desarrolladores. Esto ha contribuido a la popularidad y expansión de jQuery y ha propiciado la aparición de miles de *plugins* que nos pueden resultar de gran utilidad, ya que nos permitirán incorporar funcionalidades necesarias para nuestro sistema sin la necesidad de tener que desarrollarlas desde el principio. De este modo, el proceso de introducción de dichas funcionalidades a nuestras páginas web se convierte en un proceso de integración más que en un proceso de desarrollo, con el ahorro en tiempo (y costes) que ello conlleva.

Es importante resaltar que, a la hora de incorporar un *plugin*, se debe realizar un análisis pormenorizado de qué funcionalidades necesitamos y qué *plugin* nos las puede proporcionar, prestando atención a la calidad de ese *plugin*, su nivel de actualización y documentación e incluso a las referencias al mismo que podamos encontrar en diferentes fuentes de información más informales, como por ejemplo foros.

Puede encontrar más información sobre los *plugins* de jQuery en [1] capítulo 9, capítulo 10 y en [2].

Bibliografía Básica

1. Practical jQuery. Mukund Chaudhary, Ankur Kumar. Apress, 2015.
<http://0-proquest.safaribooksonline.com.athenea.upo.es/book/web-development/jquery/9781484207871>
2. jQuery Plugin Development Beginner's Guide, Giulio Bai, Packt Publishing Ltd, 2010.
<http://site.ebrary.com/lib/bupo/detail.action?docID=10428645&p00=jquery>



Experimentos

E1. (30 min.) Cree una página web con un formulario que contenga 3 campos para introducir el nombre, primer y segundo apellido, y un botón para enviar el formulario. Construya un *plugin* de jQuery que pinte de color azul el fondo del campo que recibe el foco del teclado. Para el campo que pierde el foco, si éste tiene contenido se volverá a poner el color del campo a blanco, y si está vacío, se pondrá de color rojo. Incluya el *plugin* en la página creada y verifique su uso.

E2. (30 min.) Usando el *plugin* AnimateScroll, cree una página donde se muestre una debajo de otra 5 imágenes en tamaño 1000x500. Al final de la página se mostrarán 5 botones que nos desplazarán a cada una de las 5 imágenes implementando cada uno de ellos un efecto distinto. El *plugin* se encuentra accesible en <http://plugins.jquery.com/animatescroll/>.

Ejercicios

EJ1. (30 min.) Usando el *plugin* oriDomi, cree una página donde se mostrarán 3 imágenes y cada una de ellas se plegará de una forma diferente. Utilice para cada una de ellas unas opciones distintas (número de paneles en horizontal en el que se plegará, número de paneles en vertical, velocidad, etc). Se valorará la personalización que se haya hecho del *plugin*. Este se encuentra accesible en <http://plugins.jquery.com/oridomi/>.

EJ2. (25 min.) Cree una página web con un formulario que solicite al usuario subir una serie de imágenes a un servidor. Para recoger dichos archivos de imagen en el formulario, use el *plugin* jQuery File Upload (<http://blueimp.github.io/jQuery-File-Upload/>).

Problemas

P1. (30 min.) Construya una página web que incluya texto e imágenes y, dentro de la cual, podremos hacer zoom (sobre cualquier elemento). Para ello, utilice el *plugin* zoomooz (<http://jaukia.github.io/zoomooz/>). Se valorará que se utilice el mayor número de parámetros de personalización de los que permite el mismo (duración, modo de escalar, etc).

P2. (60 min.) Mejore el código desarrollado en el problema 3 de la EPD 8 (hundir la flota) de manera que se habilite una zona donde arrastrar imágenes ya colocadas en el tablero de juego para eliminarlas del mismo usando el *plugin* jQuery UI Droppable (<http://plugins.jquery.com/ui.droppable/>). Además, se debe mostrar un cuadro de diálogo de jQuery UI cuando el usuario haya colocado todos los barcos en el tablero del juego informando de ello al usuario. Para incorporar estas funcionalidades haga uso de los *plugins* de jQuery UI antes mencionados (<http://jqueryui.com/demos/>).

P3. (90 min.) Modifique el código desarrollado en el problema 3 del guión de la sesión de EPD anterior (formulario de recogida de los datos personales de las familias españolas), incorporando a los datos de los pasajeros su fecha de nacimiento y realizando la validación de los campos del formulario mediante un *plugin* de jQuery. El validador comprobará que los campos obligatorios (todos menos fecha de nacimiento y número de teléfono) no se encuentran vacíos y que cada campo tiene el formato correcto. Además, incorpore otro *plugin* para darle un mejor formato de visualización al formulario, como por ejemplo Chosen (<http://plugins.jquery.com/chosen/>).

P4. (40 min.) Haciendo uso del *plugin* jQuery UI Progressbar (<http://plugins.jquery.com/ui.progressbar/>), modifique el lector de noticias desarrollado en el ejercicio 1 del guión de la EPD 8, de manera que se muestre una barra de progreso que indique el tiempo transcurrido desde que aparece una noticia hasta que ésta se cambia por la siguiente, tras lo cual la barra de progreso se reinicia. Además, la transición entre noticias se hará con el efecto *blind* de jQuery UI (desaparece desde abajo hacia arriba).

Ampliación de Bibliografía

1. Fundamentos de jQuery. Rebecca Murphey (Traducción de Leonardo D'Onofrio).
<http://librojquery.com/>
2. Professional jQuery. Cesar Otero y Rob Larsen. Wrox, 2012.
<http://site.ebrary.com/lib/bupo/detail.action?docID=10560690&p00=jquery>
3. Documentación de ayuda e iniciación al desarrollo de *plugins* de jQuery
<https://learn.jquery.com/plugins/>
4. Repositorio de *plugins* en la web oficial de jQuery
<http://plugins.jquery.com>
5. Breve tutorial de iniciación al desarrollo de *plugins* para jQuery
<http://blog.teamtreehouse.com/writing-your-own-jquery-plugins>



Datos de la Práctica

Autor del documento: Pablo Soler Regli (Diciembre 2015).

Revisiones:

1. Carlos D. Barranco (Diciembre 2015): Mejoras de redacción y formato en las secciones de objetivos, conceptos, ejercicios y problemas.
2. Ricardo – León Talavera Llamas (Diciembre 2016): Mejora en la redacción del experimento 1 y del problema 4. Se modifica el experimento 2, ejercicio 1, el problema 1 y el problema 3.
3. Carlos D. Barranco (Diciembre 2016): Mejora en el formato de objetivos, redacción de la sección conceptos y en el formato y redacción de los problemas 1, 3 y 4.
4. Gualberto Asencio Cortés (Diciembre 2017): Modificación del ejercicio 2 y problemas 2 y 4. Agregados tres nuevos enlaces de interés en la ampliación de la bibliografía. Modificado el experimento 1 para mejorar la coherencia de la aplicación.
5. Carlos D. Barranco González (Diciembre 2017): Corrección de pequeñas erratas y ajustes de formato.

Estimación temporal:

- Parte presencial: 120 minutos.
 - Explicación inicial: 5 minutos.
 - Experimentos: 60 minutos.
 - Ejercicios: 55 minutos.
- Parte no presencial: 270 minutos.
 - Lectura y estudio del guión y bibliografía básica: 50 minutos
 - Problemas: 220 minutos