

## Software y estándares para la Web

---

### **P7. JQUERY Y CONSUMO DE SERVICIOS WEB**

## Contenido

Temática del proyecto: F1 Desktop .....	2
Ejercicio 1: Introducción a AJAX utilizando jQuery .....	3
Tarea 1. Creación de un nuevo documento JavaScript.....	3
Tarea 2. Creación de la clase Fondo .....	3
Tarea 3. Creación del método de consulta a la API de Flickr.....	3
Guía para resolver la tarea 3.....	3
Tarea 4. Estableciendo la imagen de fondo .....	4
Guía para resolver la tarea 4.....	4
Resultado del ejercicio 1 .....	4
Ejercicio 2: Consulta de servicio de meteorología con jQuery .....	5
Tarea 1. Modificación del documento pais.js .....	5
Guía para resolver la tarea 1.....	5
Tarea 2. Modificación del documento meteorologia.html.....	5
Guía para resolver la tarea 2.....	6
Resultado del ejercicio 2 .....	6
Ejercicio 3: Agenda de Fórmula 1 .....	7
Tarea 1. Creación de un nuevo documento JavaScript.....	7
Tarea 2. Creación de la clase Agenda .....	7
Guía para resolver la tarea 2.....	7
Tarea 3. Llamada al servicio web .....	7
Guía para resolver la tarea 3.....	8
Tarea 4. Modificación del documento calendario.html .....	8
Guía para resolver la tarea 4.....	8
Resultado del ejercicio 3 .....	8
Recuerda .....	9

## Objetivos

En esta práctica se va a realizar:

- La creación de un documento de script utilizando el estándar ECMAScript
- La creación de objetos para la realización de diferentes tareas en combinación con HTML.
- La generación de código HTML desde el documento de script utilizando la biblioteca jQuery y su posterior inserción en el documento HTML original.
- El consumo de servicios web a través de llamadas AJAX utilizando la librería jQuery y la inserción de la información obtenida del servicio en los documentos HTML utilizando la librería jQuery.
- La validación del código HTML estático y el código HTML generado

**IMPORTANTE: Recuerda las pautas de trabajo establecidas en la primera sesión de prácticas (P0. Pautas de trabajo): valida todos los documentos HTML, valida todas las hojas de estilo CSS, comprueba la adaptabilidad y la accesibilidad con las herramientas proporcionadas.**

## Temática del proyecto: F1 Desktop

El proyecto F1 Desktop es evolutivo y será creado, completado y modificado en las diferentes prácticas de la asignatura.



## Ejercicio 1: Introducción a AJAX utilizando jQuery

jQuery es una librería escrita en JavaScript que está destinada, entre otras cuestiones, a facilitar la creación y manipulación del árbol DOM de un documento HTML.

En este ejercicio se utilizará la librería jQuery para realizar una llamada AJAX a un servicio web de imágenes para obtener una imagen que se aplique como imagen de fondo al fichero index.html del proyecto Escritorio F1.

El servicio web de imágenes recomendado para la resolución de este ejercicio es Flickr.  
<https://www.flickr.com/>

**IMPORTANTE:** Al utilizar jQuery para establecer la imagen de fondo se creará en el documento index.html un atributo style que establece el estilo correspondiente para situar la imagen como fondo de pantalla de dicho documento html. Esto se corresponde con el funcionamiento normal de jQuery y está permitido para este ejercicio.

### Tarea 1. Creación de un nuevo documento JavaScript

Dentro de la carpeta js del directorio del proyecto Escritorio F1 crea un nuevo fichero llamado fondo.js.

Añade una referencia a este nuevo fichero en el documento index.html.

Utiliza el siguiente enlace para saber cómo es la referencia que debes incluir en el fichero index.html para utilizar la opción minificada de jQuery Core 3.x: <https://releases.jquery.com/>

### Tarea 2. Creación de la clase Fondo

Dentro del fichero creado en la tarea anterior crea la clase Fondo.

Añade a esta clase un método constructor que reciba como parámetro el nombre del país, el nombre de la capital y el nombre del circuito de Formula 1 existente en el país.

Almacena estos valores en atributos dentro de la clase Fondo.

### Tarea 3. Creación del método de consulta a la API de Flickr

Crea un método que realice una consulta AJAX a la API de Flickr para obtener una imagen del circuito.

Realiza la llamada a la API de tal forma que esta conteste en formato JSON.

### Guía para resolver la tarea 3

Consulta los diferentes métodos existentes en la API de Flickr a través de este enlace:

<https://www.flickr.com/services/api/>

Identifica el método que permita obtener las fotos de una localización concreta y realiza la llamada de tal manera que devuelva una imagen asociada al circuito de F1 existente en el país.

Consulta el ejemplo incluido a continuación para conocer la forma en la que debe realizarse una llamada AJAX a través de jQuery:

<http://di002.edv.uniovi.es/~cueva/JavaScript/44jQueryJSON.html>

#### Tarea 4. Estableciendo la imagen de fondo

Utiliza una de las imágenes devueltas por el servicio de imágenes para establecerla como imagen de fondo del documento index.html

#### Guía para resolver la tarea 4

Consultar el siguiente enlace para conocer la forma en la que se puede acceder a los diferentes elementos del html utilizando la librería jQuery:

<https://learn.jquery.com/using-jquery-core/selecting-elements/>

Utiliza las fórmulas de acceso para encontrar el elemento que debe tener la imagen de fondo y establece la imagen de fondo en dicho elemento. Consulta la sección de css de jQuery para obtener información sobre como establecer la imagen de fondo:

<https://api.jquery.com/css/>

#### Resultado del ejercicio 1

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto:

- Fichero fondo.js en el directorio js del proyecto

Además, se debe haber modificado el fichero index.html para incluir la referencia al fichero JavaScript mencionado anteriormente y para incluir la llamada que permita establecer la imagen de fondo en dicho documento.

## Ejercicio 2: Consulta de servicio de meteorología con jQuery

En este ejercicio se utilizará la librería jQuery para realizar una consulta a un servicio web de meteorología que devuelva la información del tiempo en un lugar.

**El servicio web de meteorología recomendado para la resolución de este ejercicio es OpenWeatherMap.** <http://openweathermap.org/>

### Tarea 1. Modificación del documento pais.js

Modifica el documento pais.js para incluir en él el código necesario para realizar una llamada AJAX a través de jQuery al servicio web de meteorología y obtener la previsión del tiempo en la línea de meta del circuito para los próximos 5 días.

Haz que el servicio web devuelva la información del tiempo en formato XML, en idioma español y con unidades de medida del sistema métrico.

### Guía para resolver la tarea 1

Consulta los diferentes métodos existentes en la API de OpenWeatherMap a través del siguiente enlace, prestando especial atención a aquellos que pronostican el tiempo a varios días vista (forecast) y a los parámetros que reciben para su invocación:

<https://openweathermap.org/api>

Crea una cuenta gratuita en la web de OpenWeatherMap para obtener una API Key que te permita realizar peticiones a los métodos de la API existentes en el enlace anterior.

Consulta el ejemplo incluido a continuación para conocer la forma en la que debe realizarse una llamada AJAX a través de jQuery para la API de OpenWeatherMap:

<http://di002.edv.uniovi.es/~cueva/JavaScript/87-jQuery-AJAX-XML-meteo.html>

### Tarea 2. Modificación del documento meteorologia.html

Utiliza la información devuelta por el servicio web de meteorología en la tarea anterior para escribir en el documento meteorologia.html el pronóstico del tiempo para los siguientes 5 días.

Por cada día de pronóstico se debe incluir al menos la siguiente información: temperatura máxima, temperatura mínima, porcentaje de humedad, un icono que represente el tiempo que va a hacer y la cantidad de lluvia del día. Cada día de pronóstico debe estar representado por una etiqueta **article**.

Utiliza el siguiente enlace para saber cómo es la referencia que debes incluir en el fichero meteorologia.html para utilizar la opción minificada de jQuery Core 3.x: <https://releases.jquery.com/>

**IMPORTANTE:** El formato de presentación se deja libre al estudiante, valorándose la calidad de dicho formato y el uso de los diferentes conceptos de CSS. El código de estos estilos estará en el fichero `article_elements.css`

## Guía para resolver la tarea 2

Consulta el siguiente enlace para conocer los métodos existentes en jQuery que permiten crear los diferentes elementos en el documento html para representar la información.

<https://api.jquery.com/add/#post-8>

Consulta los siguientes ejemplos para ver formas en las que añadir, modificar y eliminar elementos y propiedades de elementos:

<http://di002.edv.uniovi.es/~cueva/JavaScript/80-jQuery-DOM-set-val-text-html.html>

<http://di002.edv.uniovi.es/~cueva/JavaScript/81-jQuery-DOM-set-attr.html>

<http://di002.edv.uniovi.es/~cueva/JavaScript/82-jQuery-DOM-add-elementos.html>

Modifica la hoja de estilos estilo.css para añadir en ella los estilos que sean necesarios para que los datos del tiempo se visualicen correctamente.

## Resultado del ejercicio 2

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto:

- Fichero article\_elements.css en el directorio estilo del proyecto

Además, se deben haber modificado los siguientes ficheros: el fichero pais.js para incluir el código de llamada al servicio de meteorología y el fichero meteorologia.html para incluir en él la información del pronóstico del tiempo.

## Ejercicio 3: Agenda de Fórmula 1

En este ejercicio se utilizará la librería jQuery para realizar una consulta a un servicio web de información relacionado con la Fórmula 1. Será necesario realizar una consulta que devuelva la información relacionada con las carreras de esta temporada 2024.

**El servicio web de consulta de información es la Jolpica F1, una API que sucede a Ergast F1 y cuenta con una versión compatible de todas sus llamadas <https://api.jolpi.ca/ergast/>**

### Tarea 1. Creación de un nuevo documento JavaScript

Dentro de la carpeta js del directorio del proyecto Escritorio F1 crea un nuevo fichero llamado agenda.js.

Añade una referencia a este nuevo fichero en el documento calendario.html

### Tarea 2. Creación de la clase Agenda

Dentro del fichero creado en la tarea anterior crea la clase Agenda.

Añade a esta clase un método constructor que inicialice en el objeto un atributo con la URL de la dirección que permitirá consultar la información de las carreras de la temporada en curso.

La información que devuelva la API debe estar en formato JSON.

## Guía para resolver la tarea 2

Accede a la siguiente dirección para consultar la información general de la API Ergast y conocer el método concreto para consultar la información de la temporada actual.

<http://ergast.com/mrd/>

Consulta en la dirección anterior el apartado **Race Schedule** dentro del menú **API Documentation** para conocer el método que devuelve las carreras de la temporada en curso y componer la url de llamada a la API.

### Tarea 3. Llamada al servicio web

Modifica el código del fichero agenda.js para añadir un método que permita realizar la consulta de la información sobre las carreras de la presente temporada.

Una vez obtenida la información de las carreras, el método debe recorrer dicha información y crear una estructura de elementos en el html que permita visualizar la información de cada una de las carreras de la temporada. Como mínimo se debe mostrar:

- Nombre de la carrera
- Nombre del circuito donde se celebra
- Coordenadas del circuito
- Fecha y hora de la carrera



### Guía para resolver la tarea 3

Consulta el ejemplo incluido a continuación para conocer la forma en la que debe realizarse una llamada AJAX a través de jQuery y el posterior tratamiento de la respuesta cuando la API responde en formato JSON:

<http://di002.edv.uniovi.es/~cueva/JavaScript/86-jQuery-JSON-meteo.html>

### Tarea 4. Modificación del documento calendario.html

Modifica el código del fichero calendario.html para añadir un botón que al ser pulsado solicite la información de las carreras de esta temporada a la clase Agenda y muestre esa información por pantalla.

Utiliza el siguiente enlace para saber cómo es la referencia que debes incluir en el fichero index.html para utilizar la opción minificada de jQuery Core 3.x: <https://releases.jquery.com/>

**IMPORTANTE: El formato de presentación se deja libre al estudiante, valorándose la calidad de dicho formato y el uso de los diferentes conceptos de CSS.**

### Guía para resolver la tarea 4

Consulta el ejemplo incluido a continuación para conocer la forma en la que se utilizan los botones para que ejecuten un código tras su pulsación:

<http://di002.edv.uniovi.es/~cueva/JavaScript/114-evento-botones.html>

Añade el código html necesario para visualizar la información de cada carrera utilizando los métodos existentes en jQuery para modificar el documento html. Se recomienda consultar el siguiente enlace:

<http://di002.edv.uniovi.es/~cueva/JavaScript/80-jQuery-DOM-set-val-text-html.html>

<http://di002.edv.uniovi.es/~cueva/JavaScript/81-jQuery-DOM-set-attr.html>

<http://di002.edv.uniovi.es/~cueva/JavaScript/82-jQuery-DOM-add-elementos.html>

### Resultado del ejercicio 3

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto:

- Fichero agenda.js en el directorio js del proyecto
- Fichero agenda\_grid.css en el directorio estilo del proyecto

Además, se debe haber modificado el fichero calendario.html para incluir el enlace de referencia al fichero agenda.js y el resto de código necesario para completar el ejercicio.

## Recuerda

Se requiere el uso correcto de los elementos HTML5 establecidos en los ejercicios y se valorará positivamente el uso correcto y adecuado al contexto y funcionalidad, de elementos adicionales HTML5.

Se requiere el uso correcto de las propiedades de los módulos CSS establecidos en los ejercicios y se valorará positivamente el uso correcto y adecuado al contexto, de propiedades y módulos adicionales CSS.

Solamente se permite el paradigma de orientación de objetos y todo debe estar organizado con clases y objetos. Además, no se permite el uso de ningún tipo de bibliotecas externas y debe usarse ECMAScript puro o “vanilla”, salvo en aquellos ejercicios cuyo enunciado indique que se use una biblioteca externa (por ejemplo, jQuery).

Las siguientes condiciones son de obligado cumplimiento en todo el proyecto:

- **TODOS** los documentos HTML que componen el proyecto deben ser HTML5 válidos y sin advertencias utilizando el validador de lenguajes de marcado del W3C.
  - Recuerda que el contenido de los documentos HTML puede cambiar después de la ejecución del código ECMAScript. En ese caso, se debe comprobar la validez del código HTML en todos los diferentes estados por los que pase el documento.
- Se deben utilizar las etiquetas semánticas de HTML5 (section, article, etc.) y no está permitido el uso de bloques anónimos (**div**). En aquellos ejercicios en los que se permita el uso de bloques anónimos (**div**) estará indicado en el enunciado del ejercicio; **fuera de esos ejercicios, el uso de bloques anónimos no está permitido.**
- Se deben utilizar selectores de CSS específicos, el uso de selectores id y class no está permitido. **En aquellos ejercicios en los que se permita el uso de los selectores class o id estará indicado expresamente en el enunciado del ejercicio.**
- **TODAS** las reglas de todas las hojas de estilo deben estar precedidas por un comentario donde se indique la especificidad del (o los) selectores de la regla.
- **TODAS** las hojas de estilo que se utilizan en el sitio web deben ser válidas utilizando el validador CSS del W3C
- **TODAS** las hojas de estilo deben tener 0 advertencias.
  - Recuerda seleccionar en “Más opciones” el informe de “Todas las advertencias” dentro de las opciones del validador CSS del W3C.
  - Excepcionalmente se permite las advertencias referidas a la verificación de los colores (color y background-color). **OBLIGATORIAMENTE** se debe indicar mediante un comentario en la regla de la hoja de estilo afectada la herencia de colores garantizando que la advertencia ha sido comprobada, verificada y garantizando que no provoca efectos laterales no deseados.
  - Excepcionalmente se permiten las advertencias referidas a la redefinición de propiedades derivadas del uso de @media-queries. **OBLIGATORIAMENTE** se debe indicar mediante un comentario en las reglas de la hoja de estilo afectada que propiedades se están redefiniendo.
- Se debe garantizar la adaptabilidad y realizar su verificación para todos los documentos que componen el proyecto.
  - Se deben utilizar medidas relativas en las hojas de estilo.

- Se debe garantizar la accesibilidad del proyecto mediante los test de las herramientas de accesibilidad para el nivel AAA de las WCAG 2.0 con 0 errores de modo automático en todos los documentos que lo componen.

El no cumplimiento de las características anteriores derivará en la invalidación del proyecto.