

Software y estándares para la Web

P10. COMPUTACIÓN EN EL SERVIDOR

Contenido

Temática del proyecto: MotoGP Desktop	3
Ejercicio 1: Clase Cronómetro en PHP	4
Tarea 1. Creación del documento PHP	4
Guía para resolver la tarea 1	4
Tarea 2. Creación de la clase Cronómetro	4
Guía para resolver la tarea 2	4
Tarea 3. Inicialización del cronometro.....	4
Guía para resolver la tarea 3	4
Tarea 4. Deteniendo el cronómetro	4
Guía para resolver la tarea 4	5
Tarea 5. Mostrando el tiempo transcurrido	5
Guía para resolver la tarea 5	5
Tarea 6. Interfaz para la clase Cronometro	5
Guía para resolver la tarea 6	5
Tarea 7. Prueba unitaria del cronómetro	5
Tarea 8. Validación del código dinámico del documento cronometro.php.....	5
Guía para resolver la tarea 8	6
Tarea 9. Actualización del listado de juegos en el documento juegos.html	6
Resultado del ejercicio 1.....	6
Ejercicio 2: Lectura de archivos en el servidor con PHP	7
Tarea 1. Modificación del documento clasificaciones.html.....	7
Guía para resolver la tarea 1	7
Tarea 2. Creación de la clase Clasificacion.....	7
Guía para resolver la tarea 2	7
Tarea 3. Creación del método de lectura del documento	7
Guía para resolver la tarea 3	7
Tarea 4. Mostrando el ganador de la carrera	8
Guía para resolver la tarea 4	8
Tarea 5. Mostrando la clasificación del mundial tras la carrera	8
Guía para resolver la tarea 5	8
Tarea 6. Validación del código generado.....	8
Guía para resolver la tarea 8	8
Resultado del ejercicio 2.....	8
Recuerda.....	9

Objetivos

En esta práctica se va a realizar:

- Manejo del paradigma de orientación a objetos utilizando clases en el lenguaje PHP.
- El uso del lenguaje PHP para obtener la información almacenada en archivos en el servidor web en formato XML
- La generación de código HTML usando computación en el servidor.
- La validación del código HTML generado a través de computación en el servidor.

Todos los ejemplos disponibles se pueden consultar en:

<https://web.dptoinformatica.uniovi.es/cueva/php/index.html>

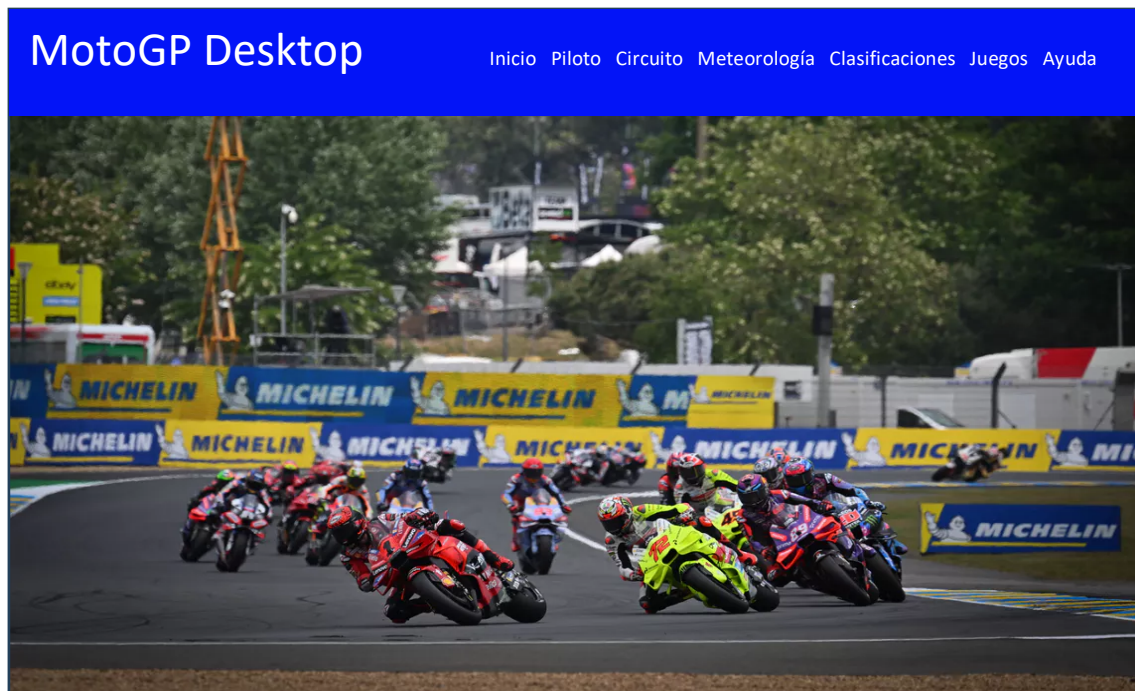
IMPORTANTE: Recuerda que los ejemplos del enlace anterior muestran el código fuente de los documentos PHP por cuestiones didácticas. Los documentos PHP del proyecto MotoGP Desktop no deben mostrar su código fuente, solamente el resultado de su ejecución en el servidor.

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.

IMPORTANTE: Está **prohibido** mezclar código JavaScript y código PHP en la realización de los ejercicios de este guion. Debe solucionarse cada ejercicio utilizando únicamente PHP y HTML.

Temática del proyecto: MotoGP Desktop

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



Ejercicio 1: Clase Cronómetro en PHP

PHP es un lenguaje de programación interpretado que se utiliza en entornos de computación en el servidor, donde la realización de las diferentes partes de computación se delega en el servidor y el código que se recibe en el cliente es el resultado de dicha computación.

En este ejercicio se creará una clase Cronómetro que permita establecer el tiempo empleado para completar una tarea en minutos, segundos y décimas de segundo.

Tarea 1. Creación del documento PHP

Dentro del directorio raíz del proyecto MotoGP-Desktop crea un nuevo documento llamado **cronometro.php**

Guía para resolver la tarea 1

Recuerda comprobar que el XAMPP está activo y con el servidor Apache arrancado, para hacer computación en el servidor es necesaria esta configuración. Para acceder a la práctica se debe hacer a través de protocolo HTTP, como en el enlace incluido a continuación:

<http://localhost/MotoGP-Desktop/cronometro.php>

Tarea 2. Creación de la clase Cronómetro

Dentro del documento creado en la tarea anterior crea la clase Cronómetro.

El constructor debe tener una instrucción de inicialice el atributo **tiempo** de la clase al valor cero.

Guía para resolver la tarea 2

Consulta el siguiente enlace para conocer cómo utilizar el paradigma de orientación a objetos en el lenguaje PHP:

<https://web.dptoinformatica.uniovi.es/cueva/php/015-Clases.php>

Tarea 3. Inicialización del cronometro

Añade un método **arrancar** a la clase Cronómetro.

Guía para resolver la tarea 3

Añade a este método la creación del atributo **inicio** de la clase Cronómetro. Este atributo marca el momento temporal en el que se inicia el cronometro.

Tarea 4. Deteniendo el cronómetro

Añade a la clase Cronómetro un método **parar** que detenga el cronómetro.

Guía para resolver la tarea 4

Utiliza la invocación al método **parar** para tomar una referencia del momento actual y calcular la cantidad de tiempo que ha transcurrido desde que se arrancó el cronómetro.

Tarea 5. Mostrando el tiempo transcurrido

Añade a la clase Cronometro un método llamado **mostrar** que muestre el cronómetro.

Guía para resolver la tarea 5

Muestra en formato mm:ss.s la cantidad de tiempo que ha contado el cronómetro desde inicio hasta el momento de su detención.

Tarea 6. Interfaz para la clase Cronometro

Dentro del documento **cronometro.php** crea la estructura general de todos los documentos que componen el proyecto MotoGP-Desktop (header con h1 y nav, migas de navegación), y enlaza las hojas de estilo `estilo.css` y `layout.css`

Crea un interfaz sencillo para realizar las pruebas de funcionamiento de la clase Cronometro. No es necesario añadir nada nuevo a nivel de CSS al proyecto MotoGP-Desktop.

Guía para resolver la tarea 6

Consulta el siguiente enlace para conocer cómo se pueden configurar los botones en HTML para que invoquen a métodos de PHP:

<https://web.dptoinformatica.uniovi.es/cueva/php/027-Botonera.php>

Añade al interfaz tres botones: uno para arrancar el cronómetro, otro para parar el cronómetro y un tercero para mostrar el tiempo transcurrido.

Modifica el código que has creado en el documento **cronometro.php** para que responda a las pulsaciones de los botones que has incluido.

Tarea 7. Prueba unitaria del cronómetro

Comprueba que se puede arrancar y parar el cronómetro correctamente y que al pulsar el botón mostrar se incluye en el interfaz la información del tiempo transcurrido entre el arranque y la parada del cronómetro.

Tarea 8. Validación del código dinámico del documento `cronometro.php`

Se debe validar el contenido del documento **cronometro.php**

Guía para resolver la tarea 8

Valida el **código dinámico** del documento **cronometro.php** usando la opción “Direct Input” del validador de HTML del W3C. Obtén el código dinámico utilizando las herramientas de desarrollador del navegador.

Tarea 9. Actualización del listado de juegos en el documento juegos.html

Actualiza la lista no ordenada de enlaces existente en el documento juegos.html y añade, en último lugar, un enlace al documento **cronometro.php** con el texto “Cronómetro PHP”.

Resultado del ejercicio 1

Una vez completado el ejercicio deberían haberse añadido los siguientes documentos al proyecto MotoGP-Desktop:

- Documento **cronometro.php** en el directorio raíz del proyecto

Además, se debe haber modificado el contenido del documento juegos.html para incluir para incluir el enlace de acceso al juego del cronometro.

Recuerda introducir la información sobre el documento **cronometro.php** en el documento de ayuda de la web.

Ejercicio 2: Lectura de archivos en el servidor con PHP

En este ejercicio se creará una clase *Clasificacion* que leerá la información existente en un documento XML en el servidor y mostrará, mediante una estructura simple con elementos HTML, parte de la información contenida en dicho documento.

Tarea 1. Modificación del documento *clasificaciones.html*

Modifica el documento *clasificaciones.html* cambiando su nombre a ***clasificaciones.php***

Guía para resolver la tarea 1

Recuerda renombrar todos los enlaces a este documento que hay en los elementos nav de todos los documentos del proyecto *MotoGP-Desktop*.

Tarea 2. Creación de la clase *Clasificacion*

Crea la clase *Clasificacion* dentro del documento ***clasificaciones.php***

Dentro del constructor de la clase anterior inicializa el atributo ***documento*** con la ruta de acceso al documento ***circuitoEsquema.xml*** que se encuentra dentro del directorio ***xml*** del proyecto *MotoGP-Desktop*.

Guía para resolver la tarea 2

Consulta el siguiente enlace para conocer cómo utilizar el paradigma de orientación a objetos en el lenguaje PHP:

<https://web.dptoinformatica.uniovi.es/cueva/php/015-Clases.php>

Tarea 3. Creación del método de lectura del documento

Crea el método ***consultar*** dentro de la clase *Clasificacion* para leer la información existente en el documento.

Guía para resolver la tarea 3

Usando el valor del atributo ***documento*** de la clase *Clasificacion* lee el contenido del archivo.

Consulta los siguientes enlaces para conocer cómo se pueden leer archivos existentes en el servidor utilizando PHP:

<https://web.dptoinformatica.uniovi.es/cueva/php/011-Leer-Archivo-Texto.php>

<https://web.dptoinformatica.uniovi.es/cueva/php/014-Leer-Archivo-XML.php>

Tarea 4. Mostrando el ganador de la carrera

Crea en el documento **clasificaciones.php** una estructuración simple con elementos de HTML para mostrar el nombre del ganador de la carrera y el tiempo empleado para ello, , contenida en el documento XML leído.

Guía para resolver la tarea 4

Consulta el enlace incluido a continuación para conocer cómo se puede iterar sobre la información leída de un documento XML con PHP:

<https://web.dptoinformatica.uniovi.es/cueva/php/014-Leer-Archivo-XML.php>

Tarea 5. Mostrando la clasificación del mundial tras la carrera

Crea en el documento **clasificaciones.php** una estructuración simple con elementos de HTML para mostrar la información de la clasificación del mundial tras la carrera, contenida en el documento XML leído.

Guía para resolver la tarea 5

Consulta el enlace incluido a continuación para conocer cómo se puede iterar sobre la información leída de un documento XML con PHP:

<https://web.dptoinformatica.uniovi.es/cueva/php/014-Leer-Archivo-XML.php>

Tarea 6. Validación del código generado

Se debe validar el contenido del documento **clasificaciones.php**

Guía para resolver la tarea 8

Valida el **código dinámico** del documento **clasificaciones.php** usando la opción “Direct Input” del validador de HTML del W3C. Obtén el código dinámico utilizando las herramientas de desarrollador del navegador.

Resultado del ejercicio 2

Una vez completado el ejercicio se debe haber modificado el nombre y el contenido del documento **clasificaciones.php** para incluir la funcionalidad del ejercicio.

Recuerda introducir la información sobre el documento **clasificaciones.php** en el documento de ayuda de la web.

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, tanto en ECMAScript como en PHP. 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 validas 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.