

Software y estándares para la Web

P6. COMPUTACIÓN EN EL CLIENTE V2

VERSIÓN 1.0 (23/10/2025) - VERSIÓN 2.0 (28/10/2025)

Contenido

Temática del proyecto: MotoGP Desktop	3
Ejercicio 1: Información de una ciudad con HTML y JS.....	4
Tarea 1. Configuración del entorno de trabajo	4
Guía para resolver la tarea 1	4
Tarea 2. Creación del documento JavaScript.....	4
Tarea 3. Creación de la clase Ciudad	4
Guía para resolver la tarea 3	4
Tarea 4. Creación de métodos en la clase Ciudad	5
Guía para resolver la tarea 4	5
Tarea 5. Instanciando la clase Ciudad para incluir información en el documento meteorologia.html.....	5
Guía para resolver la tarea 5	5
Tarea 6. Validación del código estático y generado del documento meteorologia.html	6
Guía para resolver la tarea 6	6
Resultado del ejercicio 1.....	6
Ejercicio 2: Juego de memoria con cartas - Contenido y presentación	7
Tarea 1. Creación del listado de juegos en el documento juegos.html	7
Guía para resolver la tarea 1	7
Tarea 2. Creación del documento memoria.html.....	7
Tarea 3. Creación de las cartas en memoria.html	7
Guía para resolver la tarea 3	7
Tarea 4. Creación de la presentación de las cartas.....	8
Guía para resolver la tarea 4	8
Tarea 5. Creación de la representación del juego en forma de cuadrícula de cartas.....	8
Guía para resolver la tarea 5	9
Tarea 6. Adaptabilidad del juego de cartas	9
Guía para resolver la tarea 6	9
Resultado del ejercicio 2.....	9
Ejercicio 3: Juego de memoria con cartas - Manejo de eventos.....	11
Tarea 1. Creación del documento JavaScript.....	11
Tarea 2. Creación de la clase Memoria.....	11
Tarea 3. Manejo del evento del ratón	11
Guía para resolver la tarea 3	11
Tarea 4. Instanciando la clase Memoria	12

Guía para resolver la tarea 4	12
Tarea 5. Validación del código estático y dinámico del documento memoria.html.....	12
Guía para resolver la tarea 5	12
Resultado del ejercicio 3.....	12
Recuerda.....	13
Anexo I. Circuitos del mundial de MotoGP 2025	15

Objetivos

En esta práctica se va a realizar:

- Manejo del paradigma de orientación a objetos utilizando clases en el estándar ECMAScript.
- El manejo de eventos de ratón dentro del estándar ECMAScript.
- Definición de un GRID Layout, animaciones y transiciones en el estándar CSS.
- La generación de código HTML desde el documento de script y su posterior inserción en el documento HTML original.
- 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.

IMPORTANTE: Las palabras en letra naranja negrita son nombres de variables o métodos que se deben respetar a lo largo del desarrollo de las tareas para que se pueda seguir el guion paso a paso sin problemas.

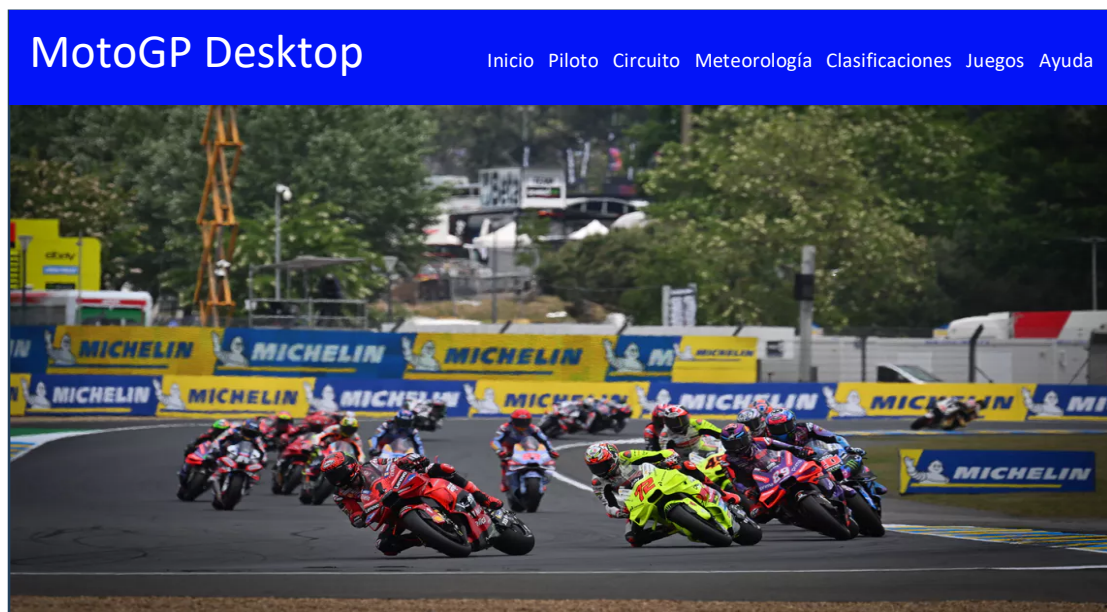
(*) **ACLARACIÓN:** Los ejemplos utilizados en los ejercicios de ECMAScript contienen código ECMAScript incrustado dentro de los archivos HTML. Esto se realiza con fines didácticos para facilitar la explicación en un único archivo. En los documentos HTML no se debe incrustar código ECMAScript, salvo: referencias a archivos ECMAScript, creación de instancias de las clases e invocación de métodos.

Todos los ejemplos disponibles se pueden consultar en:

<https://web.dptoinformatica.uniovi.es/cueva/JavaScript/CursoJavaScript.html>

Temática del proyecto: MotoGP Desktop

Se va a crear MotoGP Desktop para alojar el proyecto de la asignatura. El proyecto es evolutivo y será creado, completado y modificado en las diferentes prácticas de la asignatura.



Ejercicio 1: Información de una ciudad con HTML y JS

En este ejercicio se va a crear un archivo JS “ciudad.js” para componer un documento HTML5 con información sobre una ciudad.

La ciudad sobre la que se debe trabajar es la más próxima al circuito del mundial de MotoGP 2025 con el que se ha trabajado en las sesiones de prácticas 4 y 5 (prácticas de XML).

En el Anexo I de este documento se encuentra la tabla de circuitos del mundial de MotoGP 2025 junto con el listado de ciudades.

Tarea 1. Configuración del entorno de trabajo

A partir de este momento es necesario utilizar el servidor local Apache disponible en la herramienta XAMPP.

Guía para resolver la tarea 1

Copia y pega la carpeta del proyecto MotoGP-Desktop en la carpeta **htdocs** situada en la ruta **C://xampp/**

Abre la aplicación XAMPP y arranca el servidor web Apache pulsando en el botón **Start** del panel de control. Cuando el nombre Apache esté con el fondo verde, el servidor web estará ya disponible.

Comprueba el acceso al proyecto MotoGP-Desktop utilizando la URL correspondiente en el navegador. Por ejemplo: <http://localhost/MotoGP-Desktop>

IMPORTANTE: Recuerda que compartes la carpeta C://xampp/htdocs con cualquiera que se sienta en el mismo equipo del laboratorio L32 donde estás realizando las prácticas. Al finalizar la sesión no dejes el código de tu práctica en la máquina.

Tarea 2. Creación del documento JavaScript

Dentro de la carpeta js del directorio del proyecto MotoGP-Desktop crea un nuevo documento llamado ciudad.js

Tarea 3. Creación de la clase Ciudad

Dentro del documento creado en la tarea anterior crea la clase Ciudad y añade a dicha clase los atributos necesarios para representar la siguiente información: nombre de la ciudad, país donde se encuentra, gentilicio, cantidad de población y coordenadas de un punto central de la ciudad.

Guía para resolver la tarea 3

Se debe utilizar la ciudad vinculada al circuito que fue seleccionado en la tarea 1 de la Práctica 4 (XML y DTD) y que se encuentra en la tabla de circuitos del mundial de MotoGP 2025 del Anexo I de este documento.

Para ver cómo se crea una clase en ECMAScript puede consultarse el ejercicio:

<https://web.dptoinformatica.uniovi.es/cueva/JavaScript/68-ES6-Ejemplo-uso-de-clases.html>(*)

Tarea 4. Creación de métodos en la clase Ciudad

Añade métodos a la clase Ciudad que permitan: construir un objeto de la clase, inicializar los valores de los atributos, acceder a la información de los atributos en forma de texto y escribir información en el documento html.

Guía para resolver la tarea 4

Se deben añadir los siguientes métodos:

- Constructor: método que recibe como parámetros el nombre de la ciudad, el país y el gentilicio.
- Un método que rellene el valor del resto de atributos existentes.
- Un método que devuelva, en forma de texto, el nombre de la ciudad.
- Un método que devuelva, en forma de texto, el nombre del país.
- Un método que devuelva la información secundaria de la ciudad (gentilicio y población) con la estructura de una lista no ordenada de HTML5 dentro de una cadena.
- Un método que escriba en el documento la información de las coordenadas del punto elegido utilizando el método **document.write()**.

Observa en el siguiente ejemplo cómo se crean los métodos de una clase:

<https://web.dptoinformatica.uniovi.es/cueva/JavaScript/68-ES6-Ejemplo-uso-de-clases.html>(*)

Se aconseja consultar la especificación del método **document.write()** en:

<https://developer.mozilla.org/es/docs/Web/API/Document/write>

Tarea 5. Instanciando la clase Ciudad para incluir información en el documento meteorologia.html

Modifica el documento meteorologia.html para enlazar el documento ciudad.js y crea un objeto de la clase Ciudad con la información de la ciudad correspondiente.

Utiliza los métodos creados en la tarea anterior para escribir en el documento meteorologia.html toda la información del objeto de la clase Ciudad que has creado utilizando elementos HTML (párrafos, listas...).

Guía para resolver la tarea 5

Utiliza el elemento `<script>` para importar el documento ciudad.js como un script externo en la cabecera del documento meteorologia.html. Consulta la información sobre este elemento en los siguientes enlaces:

<https://html.spec.whatwg.org/multipage/scripting.html#the-script-element>

<https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/script>

Observa en el siguiente ejemplo cómo se crea la instancia de una clase:

<https://web.dptoinformatica.uniovi.es/cueva/JavaScript/68-ES6-Ejemplo-uso-de-clases.html>(*)

Recuerda que algunos métodos devuelven información y otros escriben directamente la información sobre el documento.

Observa el siguiente ejemplo para ver las diferentes formas en las que se puede invocar al método **write** del objeto **document** para añadir información al documento html.

<https://web.dptoinformatica.uniovi.es/cueva/JavaScript/74-InfoNavegador.html>(*)

Tarea 6. Validación del código estático y generado del documento meteorologia.html

Se debe validar el contenido del documento meteorologia.html estático y generado

Guía para resolver la tarea 6

Valida el **código estático** del documento meteorologia.html usando la opción “Validate by File Upload” del validador de HTML del W3C.

Valida el **código generado** en el documento meteorologia.html usando la opción “Direct Input” del validador de HTML del W3C. Obtén el código generado utilizando las herramientas de desarrollador del navegador.

Resultado del ejercicio 1

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

- Documento ciudad.js en el directorio js del proyecto

Además, se debe haber modificado el contenido del documento meteorologia.html para incluir la referencia a la clase Ciudad y la invocación a sus métodos.

Recuerda actualizar la información sobre el documento meteorologia.html que se incluye en el documento de ayuda de la web en base a las modificaciones realizadas en este ejercicio en ese documento.

Ejercicio 2: Juego de memoria con cartas - Contenido y presentación

En este ejercicio se utilizará GRID Layout para crear una cuadrícula de tarjetas para un juego de memoria con cartas. A lo largo de los siguientes ejercicios se irán añadiendo características hasta hacerlo completamente funcional.

En este juego de memoria con cartas habrá un grupo de 12 cartas boca abajo y el usuario deberá ir pulsando en las diferentes cartas para voltearlas y hacer parejas.

NOTA: Las imágenes para usar en las tarjetas y el código css para las tarjetas y sus animaciones y transiciones están disponibles en el campus virtual dentro de un fichero comprimido llamado cards.zip

Tarea 1. Creación del listado de juegos en el documento juegos.html

Edita el documento juegos.html y añade un elemento al inicio del documento que represente el listado de los juegos que se van a crear en el proyecto MotoGP-Desktop.

Guía para resolver la tarea 1

Utiliza una lista no ordenada de HTML para va a contener el listado de diferentes juegos en proyecto MotoGP-Desktop.

Incorpora un elemento a la lista denominado “Juego de memoria con cartas”

Tarea 2. Creación del documento memoria.html

Crea el documento memoria.html en el directorio raíz del proyecto MotoGP-Desktop. Debe tener la estructura general de todos los documentos que componen el proyecto MotoGP-Desktop (header con h1 y nav, migas de navegación), y enlazar las hojas de estilo estilo.css y layout.css

Añade al elemento de la lista de juegos creado en la tarea anterior un enlace que permita el acceso al documento memoria.html

Tarea 3. Creación de las cartas en memoria.html

Añade al documento memoria.html un título con el texto “Juego de memoria con cartas” y la estructura de elementos HTML que represente las 12 cartas del juego.

Guía para resolver la tarea 3

Añade el elemento **main** al documento memoria.html e incorpora dentro un encabezado de orden 2 para el título.

Debajo del encabezado anterior crea un elemento **article** por cada una de las tarjetas del juego. Dentro de cada elemento **article** deberá haber:

- Un encabezado de orden 3 con el texto “Memory Card”

- Una imagen

Las tarjetas del juego representarán una serie de logos relacionados con el mundial de MotoGP 2025, incluidos en la siguiente tabla. Utiliza el valor de la columna **source** como valor para el atributo **src** de la imagen y el valor de la columna **description** como valor para el atributo **alt** de la imagen.

ELEMENT	DESCRIPTION	SOURCE
Honda	Logo-Honda	multimedia/logos/honda.svg
Yamaha	Logo-Yamaha	multimedia/logos/yamaha.svg
Ducati	Logo-Ducati	multimedia/logos/ducati.svg
KTM	Logo-KTM	multimedia/logos/ktm.svg
Aprilia	Logo-Aprilia	multimedia/logos/aprilia.svg
MotoGP	Logo-MotoGP	multimedia/logos/motogp.svg

Al existir 6 elementos en la tabla anterior y necesitarse 12 cartas para el juego, se crearán dos cartas iguales para cada una de las filas de la tabla anterior.

IMPORTANTE: Los ficheros svg de las imágenes a utilizar se encuentran subidos al campus virtual dentro del fichero comprimido cards.zip y la carpeta logos de ese fichero comprimido debe descargarse dentro de la carpeta multimedia del proyecto MotoGP-Desktop

Tarea 4. Creación de la presentación de las cartas

Dentro de la carpeta estilo del proyecto crea un nuevo documento llamado cards.css

Añade una referencia a este nuevo documento de estilos en el documento memoria.html

Guía para resolver la tarea 4

El archivo comprimido **cards.zip**, disponible en el campus virtual, contiene la hoja de estilos **cards.css**, con los estilos necesarios para la presentación de las tarjetas. Copia cards.css dentro de la carpeta estilo del Proyecto MotoGP-Desktop y crea el enlace a esta hoja de estilos en el documento memoria.html

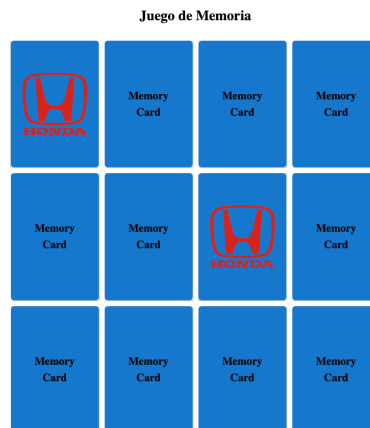
Comprueba la visualización de las tarjetas, se dispondrán verticalmente una debajo de otra siguiendo el flujo normal y no responderán a la interacción del usuario.

Tarea 5. Creación de la representación del juego en forma de cuadrícula de cartas

Dentro de la carpeta estilo del proyecto MotoGP-Desktop crea un nuevo documento llamado layout-cards.css

Añade un enlace a este nuevo documento de estilos en el documento memoria.html

Muestra las tarjetas del juego de memoria en una disposición en forma de cuadrícula de 4x3 elementos, tal y como se muestra en la imagen a continuación, utilizando un grid.



Guía para resolver la tarea 5

Aplica GRID Layout para conseguir la disposición de los elementos del juego de memoria que se observa en la imagen de ejemplo. Consulta las características del módulo [CSS-GRIDLAYOUT] a través de los siguientes enlaces:

- Referencia: <https://www.w3.org/Style/CSS/current-work>
- Más información: <https://css-tricks.com/snippets/css/complete-guide-grid/>

Debes hacer la adaptación correspondiente para que el primer elemento del elemento main (el encabezado con el título del juego) ocupe toda la primera fila por sí mismo.

IMPORTANTE: Los estilos relacionados con el grid deben incluirse en un documento layout-cards.css que solamente contendrá los estilos relacionados con el grid del juego. Modifica la cabecera del documento memoria.html para añadir la referencia al documento layout-cards.css

Tarea 6. Adaptabilidad del juego de cartas

Se debe garantizar la adaptabilidad del juego de cartas para dispositivos móviles

Guía para resolver la tarea 6

Añade una regla @media en la hoja de estilo layout-cards.css similar a la existente en layout.css donde se modifique la distribución de la cuadrícula, para que sea de 2x6 modificando los valores de las propiedades del Grid.

Resultado del ejercicio 2

Una vez completado el ejercicio 2 deberían haberse añadido los siguientes documentos al proyecto del MotoGP-Desktop:

- Documento memoria.html en el directorio raíz del proyecto
- Documento cards.css en el directorio estilo del proyecto
- Documento layout-cards.css en el directorio estilo del proyecto

Además, se debe haber modificado el contenido del documento `juegos.html` para incluir el listado de acceso a los juegos y el enlace de acceso al juego de memoria.

Recuerda actualizar la información sobre el documento `juegos.html`, que debe incluir la referencia al documento `memoria.html`, en el documento de ayuda de la web en base a las modificaciones realizadas en este ejercicio en esos documentos.

Ejercicio 3: Juego de memoria con cartas - Manejo de eventos

En este ejercicio se añadirá al juego de memoria con cartas el manejo de los eventos de ratón que permiten que el usuario interactúe con las cartas para darles la vuelta.

Las cartas de este juego de memoria pasan por dos estados diferentes durante el juego, lo que permite voltearlas cuando el usuario las descubre y dejarlas boca arriba cuando ya forman parte de una pareja descubierta por el jugador.

NOTA: Para la representación de los estados en el juego de memoria con cartas se utilizarán los atributos data de HTML5. Este tipo de atributos solo se pueden utilizar en los ejercicios que lo indiquen explícitamente y el incumplimiento de esta restricción invalidará el proyecto.

Tarea 1. Creación del documento JavaScript

Dentro de la carpeta js del directorio del proyecto MotoGP-Desktop crea un nuevo documento llamado memoria.js

Tarea 2. Creación de la clase Memoria

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

Añade a la clase Memoria un constructor que no reciba parámetros.

Tarea 3. Manejo del evento del ratón

Para que las tarjetas de memoria respondan a la interacción del usuario es necesario asociarles el evento **onclick**.

Añade a la clase Memoria un método llamado **voltearCarta** que reciba un parámetro llamado **carta**.

Añade a cada una de las tarjetas del juego un atributo **onclick** en el elemento article. Este atributo debe invocar al método **voltearCarta** pasando como parámetro la propia tarjeta utilizando la palabra reservada **this**.

El método **voltearCarta** debe añadir el atributo **data-estado** con valor **volteada** a la tarjeta que ha recibido la pulsación del usuario, que le llegará como parámetro.

Guía para resolver la tarea 3

Consultar el evento global **onclick**:

<https://developer.mozilla.org/es/docs/Web/API/GlobalEventHandlers/onclick>

Para comprender el manejo del evento **onclick** puede consultarse el ejercicio:

[https://web.dptoinformatica.uniovi.es/cueva/JavaScript/114-evento-botones.html\(*\)](https://web.dptoinformatica.uniovi.es/cueva/JavaScript/114-evento-botones.html(*))

Para comprender como se pueden manejar los valores de los atributos data puede consultarse el siguiente enlace:

<https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/dataset>

Tarea 4. Instanciando la clase Memoria

Modifica el documento memoria.html para enlazar el documento memoria.js y crea un objeto de la clase Memoria.

Comprueba que al pulsar en cada una de las tarjetas se dan la vuelta y se ven los logos de la parte cubierta de la carta.

Guía para resolver la tarea 4

Utiliza el elemento `<script>` para importar el documento memoria.js como un script externo en la cabecera del documento memoria.html. Consulta la información sobre este elemento en los siguientes enlaces:

<https://html.spec.whatwg.org/multipage/scripting.html#the-script-element>

<https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/script>

Observa en el siguiente ejemplo cómo se crea la instancia de una clase:

[https://web.dptoinformatica.uniovi.es/cueva/JavaScript/68-ES6-Ejemplo-uso-de-clases.html\(*\)](https://web.dptoinformatica.uniovi.es/cueva/JavaScript/68-ES6-Ejemplo-uso-de-clases.html(*))

Tarea 5. Validación del código estático y dinámico del documento memoria.html

Se debe validar el contenido del documento memoria.html

Guía para resolver la tarea 5

Valida el **código estático** del documento memoria.html usando la opción “Validate by File Upload” del validador de HTML del W3C.

Valida el **código dinámico** del documento memoria.html 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 3

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

- Documento memoria.js en el directorio js del proyecto

Además, se debe haber modificado el contenido del documento memoria.html para incluir la referencia a la clase Memoria y el manejo del evento onclick en las tarjetas.

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. Todo el código de las clases ECMAScript creadas para la solución de los ejercicios planteados debe estar recogido en documentos JavaScript separados, nunca incrustados en los documentos HTML. 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.

Anexo I. Circuitos del mundial de MotoGP 2025

El listado contiene los 22 circuitos del mundial de MotoGP 2025 y las ciudades asociadas.

CODIGO	CIRCUITO	CIUDAD
1	Chang International Circuit	Buriram
2	Termas de Río Hondo	Termas de Río Hondo
3	Circuit of the Americas	Austin
4	Lusail International Circuit	Lusail
5	Circuito de Jerez – Angel Nieto	Jerez de la Frontera
6	Le Mans	Le Mans
7	Silverstone	Towcester
8	MotorLand Aragon	Teruel
9	Autodromo Internazionale del Mugello	Scarperia
10	TT Circuit Assen	Assen
11	Sachsenring	Oberlungwitz
12	Automotodrom Brno	Brno
13	Red Bull Ring - Spielberg	Spielberg (Austria)
14	Balaton Park	Balaton (Hungria)
15	Circuit de Barcelona-Catalunya	Barcelona
16	Misano World Circuit Marco Simoncelli	Misano Adriático
17	Mobility Resort Motegi	Motegi
18	Pertamina Mandalika Circuit	Baturiti (Indonesia)
19	Phillip Island	Cowes (Australia)
20	Petronas Sepang International Circuit	Kuala Lumpur
21	Autódromo Internacional do Algarve	Portimao
22	Circuit Ricardo Tormo	Valencia