

Desarrollo de Front-End — Examen teórico (2 horas)

Universidad Camilo José Cela — Grado en Ingeniería Informática

Instrucciones

- Duración: **120 minutos**. Examen **individual**.
 - El uso de respuestas de Inteligencia Artificial en el examen no está permitido. En el caso de detectar respuestas de IA se suspenderá automáticamente con un cero.
 - El plagio no está permitido. Se suspenderá automáticamente con un cero.
 - Responde de forma **concisa y clara**. Cuando se pida *justificar*, una o dos frases bien dirigidas bastan.
 - La puntuación total es de **10 puntos**. Se indica el valor de cada pregunta.
 - Se debe subir un trabajo por persona en PDF indicando número de pregunta y respuesta.
 - Las preguntas erróneas NO restan.
-

Preguntas

1. **Definiciones breves** (responde en una línea cada una) [1.0 puntos]
 - a) ¿Qué es el *DOM*?
 - b) ¿Qué es *responsive design*?
 - c) ¿Qué es el *UNIX epoch* en JavaScript?
 - d) ¿Qué significa “`use strict`”?
2. **Verdadero/Falso y justifica** (si es falso, corrige la idea en una frase) [1.0 puntos]
 - a) `document.write` es una buena práctica para actualizar el DOM en 2025.
 - b) `Math.random()` devuelve enteros entre 0 y 1.
 - c) En CSS, `.rojo` y `p.rojo` tienen la misma especificidad.
 - d) En JavaScript moderno es preferible `==` a `==`.
3. **Selección DOM moderna** [0.5 puntos]
Indica dos métodos *nativos* actuales para seleccionar nodos en el DOM y una ventaja frente a técnicas obsoletas.
4. **Eventos** [0.5 puntos]
Explica con tus palabras la diferencia entre `event.target` y `event.currentTarget`, y en qué fase de propagación podrían diferir.
5. **Buenas prácticas de eventos** [0.5 puntos]
Menciona *dos* razones por las que registrar manejadores con `addEventListener` es preferible a los manejadores en línea (`onclick="..."`) en HTML.
6. **Manipulación del DOM** [0.5 puntos]
¿Cuál es la diferencia práctica entre `element.textContent = "..."` y `element.append("...")`?
7. **CSS: selectores** [0.75 puntos]
Escribe qué selecciona cada uno:
 - a) `p span`

b) `.aviso.especial`

c) `#menuPrincipal li.destacado a`

8. CSS: modelo de cajas

[0.5 puntos]

Enumera los cuatro componentes principales del *box model* y explica brevemente por qué es relevante para maquetar formularios.

9. Responsive

[0.5 puntos]

Define *mobile first* y explica el papel de la metaetiqueta *viewport* en páginas modernas.

10. Bootstrap 5 (rejilla)

[0.75 puntos]

Responde breve:

a) ¿En cuántas columnas se divide la rejilla de Bootstrap?

b) ¿Qué indica el sufijo en `.col-md-6`?

c) Escribe *dos* clases de utilidades de alineación horizontal de una fila.

11. JavaScript: tipos y comparaciones

[0.5 puntos]

¿Por qué se recomienda evitar `==` en favor de `==`? Menciona un ejemplo típico de coerción sorprendente.

12. Funciones flecha

[0.5 puntos]

Indica dos diferencias relevantes de las funciones flecha respecto a las funciones tradicionales (`function`) en JavaScript.

13. Parámetros y retorno

[0.5 puntos]

Explica brevemente *parámetros rest* y *valores por defecto*. Pon un ejemplo de una línea para cada caso (sin necesidad de que compile).

14. Manejo de errores

[0.5 puntos]

Cita dos tipos de errores estándar de JavaScript (por ejemplo, *ReferenceError*) y una buena práctica al lanzar errores propios.

15. Fecha y tiempo en JS

[0.5 puntos]

¿Qué devuelve `Date.now()` y en qué unidad? Nombra una representación “humana” recomendada para mostrar fecha/hora al usuario final.

16. Aleatoriedad

[0.5 puntos]

¿Cuándo es preferible `crypto.getRandomValues` frente a `Math.random()`? Da un ejemplo de caso de uso.

17. Módulos

[0.75 puntos]

Contrasta *ES Modules (ESM)* y *CommonJS* en una o dos frases. Añade un ejemplo de importación ESM con `default` y nombrada.

18. TypeScript (visión)

[0.5 puntos]

Explica en una o dos frases por qué TS puede mejorar la mantenibilidad en proyectos grandes de front-end.

19. Accesibilidad y SEO en formularios

[0.5 puntos]

Anota dos buenas prácticas para un formulario de *login* accesible y semántico (etiquetas o atributos concretos).

20. Criterios de calidad de una entrega

[0.5 puntos]

Cita dos indicadores de rendimiento de front-end *comunes* (nómbrelas) y por qué importan para la experiencia de usuario.