

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 no está permitido.
 - El plagio supondrá la calificación de **0**.
 - Se valorará la precisión técnica y la claridad conceptual.
 - La puntuación total es de **10 puntos**.
 - Las respuestas erróneas **NO restan**.
-

Preguntas

1. **Conceptos fundamentales** (una línea cada uno) [1.0 puntos]
 - a) Define AJAX en el contexto del desarrollo web moderno.
 - b) ¿Qué es una API REST y qué la diferencia de una API RPC?
 - c) ¿Qué es JSON Schema y para qué se utiliza?
 - d) ¿Qué se entiende por arquitectura Front-End orientada a componentes?
2. **Verdadero / Falso y justifica** [1.0 puntos]
 - a) La Same-Origin Policy impide cualquier comunicación entre dominios distintos.
 - b) `fetch()` lanza una excepción automáticamente ante códigos HTTP 4xx.
 - c) IndexedDB es síncrono y bloquea el hilo principal.
 - d) Un Service Worker puede interceptar peticiones de red.
3. **Modelo cliente-servidor** [0.75 puntos]

Describe el flujo completo de una petición AJAX desde que el usuario interactúa con la interfaz hasta que el DOM se actualiza.
4. **Asincronía en JavaScript** [0.75 puntos]

Explica por qué el uso de `async/await` mejora la legibilidad del código frente al encadenamiento de promesas.
5. **HTTP y REST** [0.75 puntos]

Indica la diferencia entre los métodos GET, POST y PUT, y da un ejemplo razonable de uso para cada uno.
6. **JSON: diseño de datos** [0.75 puntos]

Enumera dos buenas prácticas en el diseño de estructuras JSON para APIs públicas y justifica brevemente cada una.
7. **Express.js** [0.75 puntos]

Explica la diferencia entre:

 - a) `req.params` y `req.body`
 - b) `res.send()` y `res.json()`

8. **Arquitectura MVP** [0.75 puntos]
Justifica por qué el patrón MVP mejora la mantenibilidad y testabilidad en aplicaciones Front-End complejas.
9. **APIs externas** [0.75 puntos]
¿Qué riesgos técnicos y legales pueden aparecer al integrar APIs de terceros en una aplicación web?
10. **APIs HTML5 y contexto seguro** [0.75 puntos]
Explica qué es un contexto seguro (`https` o `localhost`) y nombra dos APIs que lo requieran obligatoriamente.
11. **Service Workers** [0.75 puntos]
Describe el ciclo de vida básico de un Service Worker e indica una ventaja clara para la experiencia de usuario.
12. **Almacenamiento en el navegador** [0.75 puntos]
Compara `localStorage` e `IndexedDB` en términos de:
- a) Capacidad
 - b) Casos de uso
13. **Rendimiento Front-End** [0.75 puntos]
Explica dos técnicas para mejorar el rendimiento percibido de una aplicación web moderna.
14. **Integración avanzada** [0.75 puntos]
¿Por qué es importante manejar correctamente los errores de red y estados HTTP al consumir una API REST?
15. **Práctica final del curso** [0.75 puntos]
Relaciona AJAX, JSON y arquitectura MVP en el contexto de la ampliación final del e-commerce desarrollada durante el curso.