

EXAMEN

Examen de Validación: Desarrollo Web e Intercambio de Datos

Nombre: _____ Código: _____ Fecha: _____

1. Sobre Innovación y UI (Basado en Tendencias y Diseño)

Si queremos que la aplicación de salud **VitalStream** no parezca un formulario aburrido, ¿cuál de las siguientes técnicas de diseño actuales ayudaría a mejorar la **jerarquía visual** y la interacción sin saturar al usuario?

- a) Usar tablas HTML clásicas con bordes gruesos para separar datos.
- b) Implementar **Micro-interacciones** (transiciones suaves) y el uso de **Espacios en Blanco** para reducir la carga cognitiva.
- c) Llenar la pantalla con texto explicativo para que el usuario no se pierda.
- d) Usar solo colores primarios brillantes (rojo, azul, amarillo) para llamar la atención.

2. Sobre HTML Semántico

En el video de HTML Semántico se recalca que las etiquetas no solo sirven para "ver" la web. Si estás maquetando el formulario de síntomas de un paciente, ¿por qué es fundamental usar etiquetas como `<label>`, `<fieldset>` y `<button>` en lugar de simples `<div>` o ``?

- a) Porque el código se ve más corto.
- b) Por **accesibilidad** (para que lectores de pantalla entiendan el formulario) y para mejorar el SEO.
- c) Porque el navegador no permite usar `<div>` dentro de un formulario.
- d) Para que el archivo CSS sea más fácil de escribir.

3. Sobre Layouts con CSS (Grid y Flexbox)

Según lo aprendido, ¿cuál sería la combinación técnica más eficiente para el diseño de la App?

- a) Usar float: left para todas las columnas de la aplicación.
- b) Usar **CSS Grid** para la estructura general de la página (layout) y **Flexbox** para alinear los elementos internos (iconos y textos) dentro de las tarjetas de síntomas.
- c) Usar tablas de HTML para posicionar los botones de envío.

- d) Solo se debe usar Flexbox, ya que Grid es incompatible con dispositivos móviles.

4. Sobre el Protocolo HTTP y el Paso de Datos

Cuando el paciente termina de llenar sus signos vitales y presiona el botón de "Enviar", se realiza una petición al servidor. Según el video de HTTP, ¿cuál es el **método** correcto y qué **código de estado** esperarías si los datos se guardaron exitosamente?

- a) Método GET y código 404 (Not Found).
- b) Método **POST** y código **201 (Created)** o **200 (OK)**.
- c) Método DELETE y código 500 (Internal Server Error).
- d) Método POST y código 403 (Forbidden).

5. Sobre el Formato de Intercambio (JSON)

Si vas a enviar la temperatura (38.5) y el ritmo cardíaco (80) de un paciente al servidor mediante una API Fetch, ¿cuál es el formato de objeto **JSON** correctamente estructurado?

- a) <datos><temp>38.5</temp><ritmo>80</ritmo></datos>
- b) temperatura=38.5&ritmo=80
- c) { "temperatura": 38.5, "ritmo": 80 }
- d) [temperatura: 38.5; ritmo: 80]