

3. Al finalizar realizar en un documento aparte lo siguiente.

a. Una tabla que contenga lo siguiente:

Tipo de evento	Elemento afectado	Resumen de lo que hace
click	Botón	Muestra una alerta cuando se presiona
input	Campo de texto	Muestra en consola lo que el usuario escribe
mouseover	Caja	Cambia el color cuando el mouse pasa encima
mouseout	Caja	Regresa el color cuando el mouse sale
change	Select	Cambia el color del fondo según selección
submit	Formulario	Evita recarga y muestra mensaje
keydown	Input numérico	Incrementa contador al presionar tecla
dblclick	Botón	Cambia color al hacer doble clic

b. Contestar las siguientes preguntas:

- ¿Utilizó listeners para sus eventos o los describió como propiedades de los elementos? Justifique su elección

R// Se utilizaron **event listeners** mediante el método `addEventListener()`.

Esta elección permite:

- Separar el HTML del JavaScript.
- Agregar múltiples eventos al mismo elemento.
- Mantener un código más organizado y profesional.
- Facilitar mantenimiento y escalabilidad del proyecto.

Por estas razones, los listeners son una mejor práctica en comparación con describir eventos como propiedades (`onclick`, `onchange`, etc.).

- Seleccione uno de sus eventos y explique porque ese tipo de evento era la mejor elección para esa situación y que otro podría haber utilizado

R// Se seleccionó el evento **submit** en el formulario.

El evento submit es la mejor opción porque:

- Se activa cuando el usuario intenta enviar el formulario.
- Permite validar información antes de enviarla.
- Permite usar `preventDefault()` para evitar la recarga automática de la página.

Otro evento que podría haberse utilizado es click en el botón de enviar, pero no sería la mejor opción porque no captura el envío completo del formulario, especialmente si el usuario presiona Enter.

Por lo tanto, submit es más adecuado y profesional.