

**INSTRUCCIONES:**

- El examen debe entregarse a través del enlace habilitado en el aula virtual en un archivo ZIP con el nombre **Examen\_Unidad4\_Nombre\_Apellido1**. Por ejemplo, si tu nombre es José López Pérez, el archivo se denominará Examen\_Unidad4\_José\_López.
- Los archivos de cada ejercicio deben incluir también tu nombre y apellidos. Por ejemplo, **Ejercicio1\_Nombre\_Apellido1.html** (Ejercicio1\_José\_López.html).
- Para algunos ejercicios, dado que es necesario implementar secuencias que implican movimiento, hay dos vídeos en el archivo con el enunciado del examen denominados **Ejercicio1.mp4** y **Ejercicio3.mp4** que muestran el comportamiento de la página.

**Ejercicio 1 (4 puntos)**

Modifica la página web Ejercicio 1.html de la carpeta “Ejercicio 1”, teniendo en cuenta las siguientes consideraciones:

- Para facilitar la realización del ejercicio, no es necesario que muevas el código de JavaScript a un archivo .js (aunque recuerda que en una página web real es lo más adecuado).
- El ejercicio se puede resolver sin cambiar nada del código HTML y CSS (este último, se ha incluido en un fichero aparte por este mismo motivo). No obstante, observar el código con detenimiento te puede resultar de gran ayuda.
- Abre la página web y echa un vistazo al código HTML. Hay tres elementos diferenciados: una imagen con un balón, un cuadrado rojo al que llamaremos pulsador y un campo de texto en el que introduciremos un número de segundos que duran las animaciones.



- Al hacer click en el pulsador, se generan las siguientes secuencias:
  1. Desaparece el pulsador inmediatamente.
  2. Se generan los dos efectos que se indican a continuación simultáneamente:
    - **Desplazar la imagen un ancho de 70% a la derecha** en “n” segundos según el campo de texto.
    - **Rotar la imagen 360 grados** en “n” segundos según el campo de texto.
  3. Vuelve a aparecer el pulsador, pero solo cuando han terminado los efectos anteriores.
- Cuando finaliza la secuencia, la imagen se queda en la posición final y solamente volverá a la posición inicial si hacemos click de nuevo en el pulsador.
- Por último, si el usuario no ha introducido ningún valor en el campo de texto o es menor igual que cero, se mostrará un alert con el texto “Debe introducir un número de segundos mayor que cero”.
- El ejercicio debe emplear promesas de JavaScript al menos una vez. Si no consigues que la secuencia se genere en orden, no te atasques. Por ejemplo, el ejercicio puntuará parcialmente si consigues el desplazamiento y rotación de la imagen, aunque no desaparezca el pulsador.
- En el vídeo **Ejercicio1.mp4** podrás ver más claramente el resultado que se pretende.

## Ejercicio 2 (3 puntos)

Crea una carpeta denominada Ejercicio 2. A continuación, elabora una nueva etiqueta basada en un componente web que se denominará **mi-semaforo** y cuya visualización sea similar a la siguiente imagen.



Además, deberás tener en cuenta las siguientes consideraciones:

- El código HTML y CSS se incluirá empleando templates y shadow DOM.
- El componente web tiene un atributo de HTML opcional denominado “seconds”. Debes preparar el código del componente para acceder directamente a este atributo con [this.seconds](#).
- Se recomienda implementar los colores de fondo gris, rojo y verde con clases de CSS, ya que facilitará el posterior desarrollo del ejercicio 3.

## Ejercicio 3 (3 puntos)

Ahora vamos a añadir funcionalidad al componente creado en el ejercicio 2. Para ello, puedes copiar el código anterior a una carpeta denominada Ejercicio 3, o resolver ambos ejercicios en una carpeta denominada Ejercicio 2-3.

[Si no consigues resolver el ejercicio anterior, puedes implementar las acciones de este ejercicio sin componentes web y puntuará la parte proporcional.](#)

El componente responderá con las acciones que se indican debajo al hacer click en cada uno de los recuadros:

- Para algunas acciones, el número de segundos indicado se basa en el atributo seconds.
- [Si el usuario hace click en el cuadrado de arriba, se cambia el color de este cuadrado a rojo y el cuadrado de debajo a gris. Además, se para la animación del cuadrado de debajo si se hubiera iniciado \(se explica a continuación\). Estas acciones se realizan inmediatamente sin tener en cuenta el atributo seconds.](#)
- [Si el usuario hace click en el cuadrado situado debajo, se cambia el color del cuadrado de arriba a gris. Además, se inicia una animación que cambia de manera intermitente el color de este cuadrado de verde a gris cada cierto tiempo equivalente a una cantidad de segundos del atributo seconds \(si no se especifica, se toma un valor por defecto de 1 segundo\).](#)

Por otro lado, el componente web detectará si se ha añadido un nuevo valor para el atributo seconds. En ese caso se vuelve a la visualización inicial del ejercicio 2 (cuadrado de arriba en rojo y de debajo en gris) y se para la animación del cuadrado inferior si se hubiera iniciado.

Para probar que se inicializa el componente al cambiar el atributo `seconds`, puedes añadir un botón o un elemento que accede desde el HTML principal a la etiqueta creada con un componente web y cambia dinámicamente este atributo. Por ejemplo, `document.querySelector("mi-semaforo").setAttribute(...)` o `document.querySelector("mi-semaforo").seconds = ...`

En el vídeo [Ejercicio3.mp4](#) podrás ver más claramente el resultado que se pretende:

- Primero se hacen varias pruebas con el atributo `seconds` directamente a través del HTML con el valor de 0,5 segundos.
- En segundo lugar, el botón “Reiniciar”, que se añade debajo del componente, se encarga de cambiar el atributo `seconds` a 1 segundo y volver al estado inicial parando la animación del cuadrado de debajo si se hubiera iniciado. Posteriormente se vuelven a hacer una serie de pruebas para comprobar que ahora se aplica una cantidad de segundos diferente.

Por último, como se indicaba en el ejercicio 2, se recomienda trabajar con clases y los atributos `className` o `classList` de JavaScript para cambiar de color cada cuadrado.