

### 3. Documento README

a. Tabla de selectores utilizados en la página web:

Tipo de selector	Elemento afectado	Descripción de la tarea que realiza el elemento
getElementById	div#color-box	Cambia el color de fondo del div al pulsar cada botón de color.
querySelector	p#counter-num	Selecciona el párrafo del contador y actualiza su valor numérico (+/-/reset).
querySelectorAll	li.item-pill	Obtiene todos los ítems de la lista y permite activar o desactivar cada etiqueta tecnológica.
getElementsByTagName	p (dentro de #paragraph-area)	Selecciona todos los párrafos del área y cambia su tamaño de fuente y color.
getElementsByClassName	span.highlight-able	Selecciona todos los spans con esa clase y aplica o quita el resaltado amarillo.
querySelector + input	input#name-input	Detecta el evento input en tiempo real y actualiza el saludo en el output-box.
getElementById + classList.toggle	img#hero-img	Alterna la clase CSS .colored para cambiar la imagen entre blanco/negro y color.
querySelectorAll (DOM)	tr dentro de #table-body	Agrega o elimina filas en la tabla dinámicamente y actualiza el conteo de filas.

#### b. Preguntas:

##### ¿Cuál es la ventaja de utilizar getElementById frente al querySelector?

getElementById es el método más rápido para acceder a un elemento cuando se conoce su atributo id. El navegador mantiene un índice interno de todos los IDs del documento, por lo que la búsqueda se realiza en tiempo constante  $O(1)$ , sin necesidad de recorrer el árbol DOM.

En cambio, querySelector debe analizar la cadena del selector CSS y recorrer el árbol DOM para encontrar el primer nodo que coincida, lo cual es un proceso algo más costoso. Además, querySelector acepta cualquier selector CSS válido (clases, atributos, pseudoclases, combinaciones), mientras que getElementById solo funciona con el valor exacto del atributo id.

Por lo tanto: si el elemento tiene un id definido y se necesita acceder a él de forma directa y eficiente, getElementById es la mejor opción. Si se necesita un selector más complejo o condicional, querySelector es más flexible.

##### ¿En qué situación es preferible usar querySelectorAll en lugar de querySelector? Explique qué tipo de dato devuelve cada uno.

Tipos de dato que devuelve cada método:

- **querySelector** → devuelve un único Element (el primero que coincide con el selector CSS) o null si no existe ninguno.

- **querySelectorAll** → devuelve un NodeList estático con todos los elementos que coinciden con el selector. Es una "fotografía" del DOM en el momento de la llamada; no se actualiza si el DOM cambia después.

**Es preferible usar querySelectorAll cuando:**

- Se necesita manipular o iterar sobre múltiples elementos al mismo tiempo (ejemplo: todos los botones, todos los párrafos, todas las celdas de una tabla).
- Se requiere aplicar un estilo, clase o evento a un conjunto de nodos que comparten una clase o atributo CSS.
- Se quiere contar cuántos elementos coinciden antes de procesarlos, usando la propiedad `.length` del NodeList.