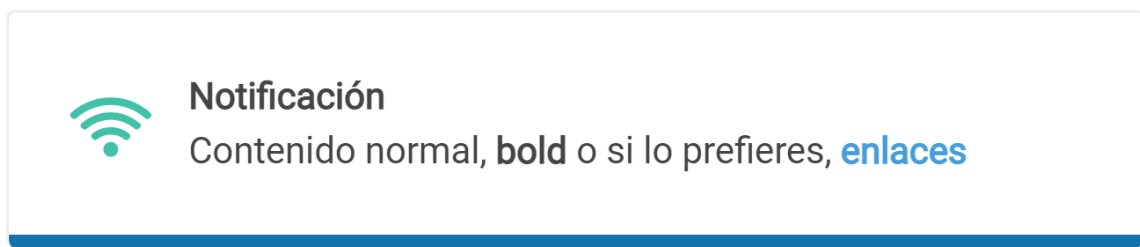


# Toast Notification

## Versión 1

Probablemente alguna vez te viste en la necesidad de generar notificaciones flotantes para informar al usuario sobre ciertas acciones o procesos completados, pero al buscar en internet no has encontrado lo que buscabas o en el peor de los casos, lo encontraste y no pudiste modificarlo porque el código es tan difícil que editar su comportamiento se vuelve una tarea compleja que simplemente decidiste no utilizarlo. Para ello, he puesto manos a la obra creando esta clase JavaScript que te ayudara con esta tarea sencilla, pero a la vez, confusa si nunca lo has utilizado. Te explicaré paso por paso como es que se genera, que estilos se aplican y lo mejor, eres libre de utilizarlo y modificarlo para adaptarlo mejor a tu proyecto.



*Ejemplo de notificación generada.*

En el documento HTML deberá existir un contenedor con el id “notification-container”:

```
1 <div class="notification-container unselectable" id="notification-container">
2 </div>
3 <script src="js/toast_notification.js">
4 </script>
```

La constante “CONTAINER\_QUOTE” almacena este elemento. Es necesario para poder mostrar las notificaciones.

```
1 const CONTAINER_QUOTE = document.getElementById('notification-container');
```

La clase “createToastMessage” crea una nueva notificación, su constructor recibe un objeto de configuración que requiere “iconClass”, “title”, “body”, “color” y “displayTime”, su uso es el siguiente:

**iconClass:** (Requerido) Especifica una clase CSS que muestre un icono. Trabaja con “background-image”.



Titulo de la notificación

Contenido

*Icono de la notificación insertado por medio de sprite.*

**title:** (Requerido) Especifica el título de la notificación.



Titulo de la notificación

Contenido

*Titulo de la notificación: Título de la notificación.*

**body:** (Requerido) Especifica el cuerpo de la notificación, se inserta por medio de “innerHTML” en JavaScript por lo que puede contener etiquetas HTML dentro. Por defecto, las etiquetas “p”, “span” y “a” tienen estilos CSS, para agregar más, es necesario declarar sus propiedades CSS.

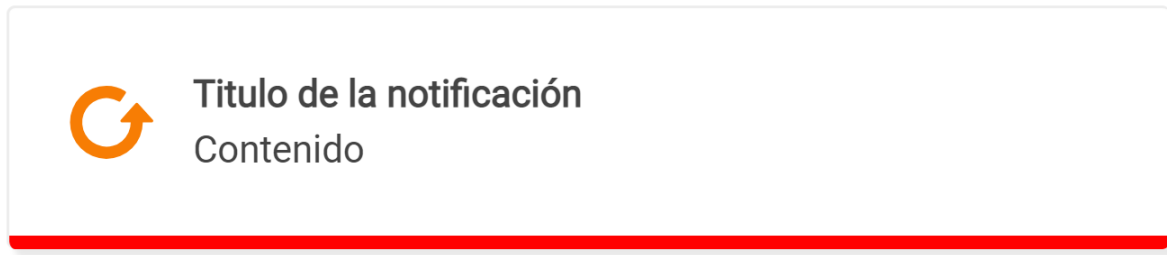


Titulo de la notificación

Contenido

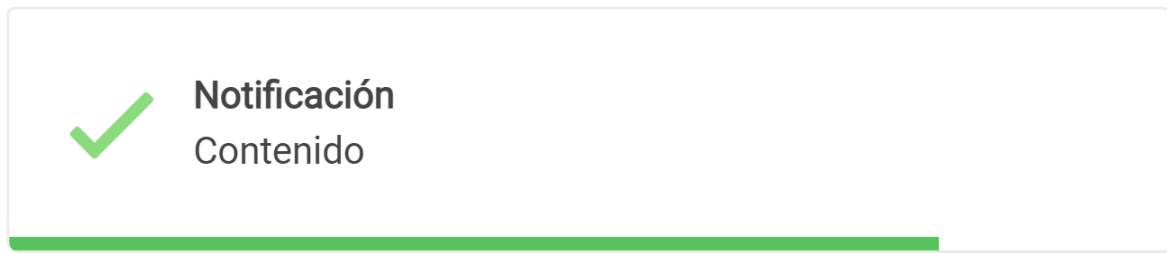
*Contenido de la notificación: “<p>Contenido</p>”.*

**color:** (Opcional) Especifica el color de la barra de progreso en una notificación. Si no se especifica, por defecto el color en hexadecimal será #64B8D3.



*Notificación con barra de progreso color "red".*

**displayTime:** (Opcional) Especifica el tiempo de vida en milisegundos de una notificación. Si no se especifica, por defecto la notificación no desaparecerá a menos que el usuario pulse sobre ella para eliminarla.



*Notificación al 80% de su tiempo de vida.*

Sabiendo esto, el objeto de configuración debe estar formado de la siguiente manera:

```
1  const config = {  
2    iconClass: "notification-tab__icon--check",  
3    title: "Título de la notificación",  
4    body: "<p>Contenido normal, <span>bold</span> o si lo prefieres, <a href='#'>enlaces</a>",  
5    color: "#647CD3",  
6    displayTime: 5000  
7  };
```

*Ejemplo de objeto de configuración utilizado por el constructor para generar una notificación.*

*La propiedad "color" y "displayTime" son opcionales, pero causaran un comportamiento como se mencionó anteriormente.*

El constructor recibe los datos del objeto de configuración y los asigna al objeto de la notificación, en caso de no encontrar alguno, agregara valores por defecto.

Nota: a pesar de que los valores se pueden asignar por defecto, es recomendable especificarlos en el objeto de configuración o de lo contrario, se obtendrá una notificación genérica sin valor alguno.

Una vez que el constructor ha asignado los valores, llamara a la función “eventGroup\_CreateToastNotification”, la cual es un grupo de funciones necesarias para crear una notificación.

Despues verificará si la variable “displayTime” tiene un valor definido o no.

Cuando tiene un valor definido: Agregara un temporizador a la notificación y llamara a las funciones “change\_Timer\_When\_Mouseout” y “change\_Timer\_When\_Mouseover” por medio de un grupo de funciones de nombre “eventGroup\_NotificationTimer”, esto provoca que el temporizador se pause al pasar el cursor por encima de la notificación y al dejar de estar sobre ella, el temporizador se reiniciara.

Cuando no tiene un valor definido: No Agregará las funciones de temporizador y la notificación no será eliminada por cuenta propia.

Finalmente, se llamará a la función “remove\_ToastNotification\_When\_Click” para que, al presionar sobre la notificación, se elimine.

```
1  constructor(config){
2
3      // Add validation here (Required)
4
5      this.iconClass = config.iconClass || "notification-tab__icon--information";
6      this.title = config.title || "Título";
7      this.body = config.body || "<p>Contenido</p>";
8      this.color = config.color || "#64B8D3";
9      this.displayTime = config.displayTime || null;
10
11     this.eventGroup_CreateToastNotification();
12
13     if(this.displayTime){
14         this.time_original = this.displayTime;
15         this.time_remain = this.displayTime;
16         this.time_percentage = 100;
17         this.eventGroup_NotificationTimer();
18     }
19
20     this.remove_ToastNotification_When_Click();
21 }
```

*Constructor de la clase “createToastMessage”.*

## Creación de una notificación

La creación de una notificación esta dividida en varias funciones, pero puedes llamar al grupo de funciones “eventGroup\_CreateToastNotification”.

```
1 eventGroup_CreateToastNotification(){
2     this.createTabElements();
3     this.addClassToTabElements();
4     this.addContentToElements();
5     this.joinTabToDocument();
6 }
```

*Grupo de funciones para crear una notificación.*

La función “createTabElements” creará las etiquetas HTML necesarias para la notificación por medio de JS.

```
1 createTabElements(){
2     // notification-tab
3     this.notification_tab = document.createElement('div');
4
5     // notification-tab__content-container
6     this.notification_tab__content_container = document.createElement('div');
7
8     // notification-tab__icon-container
9     this.notification_tab__icon_container = document.createElement('div');
10
11    // notification-tab__icon
12    this.notification_tab__icon = document.createElement('div');
13
14    // notification-tab__text-container
15    this.notification_tab__text_container = document.createElement('div');
16
17    // notification-tab__title
18    this.notification_tab__title = document.createElement('div');
19
20    // notification-tab__title (etiqueta <p>)
21    this.notification_tab__title_content = document.createElement('p');
22
23    // notification-tab__description
24    this.notification_tab__description = document.createElement('div');
25
26    // notification-tab__progressbar-container
27    this.notification_tab__progressbar_container = document.createElement('div');
28
29    // notification-tab__progressbar
30    this.notification_tab__progressbar = document.createElement('span');
31 }
```

*Función responsable de generar las etiquetas HTML de una notificación.*

La función “addClassToTabElements” asignara clases CSS a los elementos HTML creados con la función “createTabElements”.

```
1 addClassToTabElements(){
2   this.notification_tab.classList.add('notification-tab');
3   this.notification_tab__content_container.classList.add('notification-tab__content-container');
4   this.notification_tab__icon_container.classList.add('notification-tab__icon-container');
5   this.notification_tab__icon.classList.add('notification-tab__icon');
6   this.notification_tab__text_container.classList.add('notification-tab__text-container');
7   this.notification_tab__title.classList.add('notification-tab__title');
8   this.notification_tab__description.classList.add('notification-tab__description');
9   this.notification_tab__progressbar_container.classList.add('notification-tab__progressbar-container');
10  this.notification_tab__progressbar.classList.add('notification-tab__progressbar');
11 }
```

*Asignación de clases CSS para dar estilos a una notificación.*

La función “addContentToElements” agrega el contenido a los elementos HTML haciendo uso de los valores definidos en el objeto de configuración y almacenados en el objeto de notificación.

```
1 addContentToElements(){
2   // set icon (class)
3   this.notification_tab__icon.classList.add(this.iconClass);
4
5   // title
6   this.notification_tab__title_content.innerHTML = this.title;
7
8   // body
9   this.notification_tab__description.innerHTML = this.body;
10
11  // color
12  this.notification_tab__progressbar.style.background = this.color;
13 }
```

*Asignación de contenido para una notificación.*

La función “joinTabToDocument” tiene dos pasos, primero generará la estructura HTML de una notificación siguiendo la estructura de una notificación:

```
1 <div class="notification-tab notification-tab--initial-animation">
2   <div class="notification-tab__content-container">
3     <div class="notification-tab__icon-container">
4       <div class="notification-tab__icon config.iconClass">
5       </div>
6     </div>
7     <div class="notification-tab__text-container">
8       <div class="notification-tab__title">
9         config.title
10      </div>
11      <div class="notification-tab__description">
12        config.body
13      </div>
14    </div>
15  </div>
16  <div class="notification-tab__progressbar-container">
17    <span class="notification-tab__progressbar" style="background: config.color;"></span>
18  </div>
19 </div>
```

*Estructura de una notificación.*

Después insertará la notificación en la constante “CONTAINER\_QUOTE” que, como se mencionó anteriormente, es donde se insertarán las notificaciones generadas. Finalmente fuerza el repintado con “clientHeight” para poder mostrar las animaciones en la notificación que fue creada dinámicamente. Adicionalmente se agrega la clase CSS “notification-tab—initial-animation” para ejecutar una animación de entrada a la notificación.

```
1 joinTabToDocument(){
2     // Add notification-tab__progressbar to notification-tab__progressbar-container (Group 1)
3     this.notification_tab__progressbar_container.appendChild(this.notification_tab__progressbar);
4
5     // Add notification-tab__icon to notification-tab__icon-container (Group 2)
6     this.notification_tab__icon_container.appendChild(this.notification_tab__icon);
7
8     // Add <p> to notification-tab__title (Group 3)
9     this.notification_tab__title.appendChild(this.notification_tab__title_content);
10
11    // Add notification-tab__title to notification-tab__text-container
12    this.notification_tab__text_container.appendChild(this.notification_tab__title);
13
14    // Add notification-tab__description to notification-tab__text-container
15    this.notification_tab__text_container.appendChild(this.notification_tab__description);
16
17    // Add notification-tab__icon-container to notification-tab__content-container
18    this.notification_tab__content_container.appendChild(this.notification_tab__icon_container);
19
20    // Add notification-tab__text-container to notification-tab__content-container
21    this.notification_tab__content_container.appendChild(this.notification_tab__text_container);
22
23    // Add notification-tab__content-container to notification-tab
24    this.notification_tab.appendChild(this.notification_tab__content_container);
25
26    // Add notification-tab__progressbar-container to notification-tab
27    this.notification_tab.appendChild(this.notification_tab__progressbar_container);
28
29    // Add Layout 1 to queue
30    CONTAINER_QUOTE.appendChild(this.notification_tab);
31
32    // Paint (element) on queue
33    this.notification_tab.clientHeight;
34
35    // Add initial animation
36    this.notification_tab.classList.add('notification-tab--initial-animation');
37 }
```

*Función que mostrará la notificación generada.*

## Temporizador de una notificación

El temporizador esta dividido en varias funciones y solo puede ser llamado si el objeto de configuración definió la propiedad “displayTime”, solo así se llamara al grupo de funciones de nombre “eventGroup\_NotificationTimer”.

```
1  eventGroup_NotificationTimer(){
2      this.set_Timer_For_ToastNotification();
3      this.change_Timer_When_Mouseout();
4      this.change_Timer_When_Mouseover();
5  }
```

*Grupo de funciones para el temporizador de una notificación.*

La función “set\_Timer\_For\_ToastNotification” genera un temporizador que avanza cada 100 milisegundos, restando el tiempo de vida de una notificación según lo definido en el objeto de configuración, el progreso afectara a la barra de progreso de la notificación. Una vez llegado a su fin, la notificación será eliminada llamando a la función “remove\_ToastNotification”.

```
1  set_Timer_For_ToastNotification(){
2      this.displayTimeCounter = setInterval(() => {
3          this.time_remain -= 100;
4          // calculate percentage
5          this.time_percentage = ((100 / this.time_original) * this.time_remain);
6          this.notification_tab__progressbar.style.width = `${this.time_percentage}%`;
7
8          // Stop timer
9          if(this.time_remain === 0 || this.time_remain <= 0){
10             clearInterval(this.displayTimeCounter);
11             this.remove_ToastNotification();
12         }
13     }, 100);
14 }
```

*Temporizador para una notificación.*



La función “change\_Timer\_When\_Mouseover” restablecerá el tiempo de vida de la notificación y eliminará el temporizador, también afecta a la barra de progreso que volverá a ser de 100%.

```
1 change_Timer_When_Mouseover(){
2     this.notification_tab.addEventListener('mouseover', ()=>{
3         this.time_remain = this.time_original;
4         this.time_percentage = 100;
5         this.notification_tab__progressbar.style.width = `${this.time_percentage}%`;
6         clearInterval(this.displayTimeCounter);
7     });
8 }
```

*Función que restablece el temporizador cuando el cursor se encuentra en la notificación.*

La función “change\_Timer\_When\_Mouseout” vuelve a establecer e iniciar el temporizador cuando el cursor ya no se encuentra sobre la notificación.

```
1 change_Timer_When_Mouseout(){
2     this.notification_tab.addEventListener('mouseout', () => {
3         this.set_Timer_For_ToastNotification();
4     });
5 }
```

*Función que vuelve a establecer un temporizador cuando el cursor abandona la notificación.*

## Eliminar una notificación

Existen dos formas de eliminar una notificación, la principal y mas utilizada es al finalizar su tiempo de vida establecido en el objeto de configuración, pero también se puede eliminar al pulsar sobre ella.

La función “remove\_ToastNotification” elimina el temporizador (en caso de que exista), agrega clases CSS a elementos de la notificación para ejecutar una animación de salida y al transcurrir 250 milisegundos, la notificación y su objeto, serán eliminados.

```
1  remove_ToastNotification(){
2      // If timer exists
3      if(this.displayTimeCounter){
4          clearInterval(this.displayTimeCounter);
5      }
6
7      // Add class to animation
8      this.notification_tab.classList.add('notification-tab--delete-tab');
9      this.notification_tab__title.classList.add('notification-tab--delete-text');
10     this.notification_tab__description.classList.add('notification-tab--delete-text');
11     this.notification_tab__icon.classList.add('notification-tab--delete-icon');
12
13     // Time out (Animation)
14     this.timeToCleanUp = setTimeout(() => {
15         this.notification_tab.remove();
16         clearTimeout(this.timeToCleanUp);
17     }, 250);
18
19     delete this.createToastMessage;
20 }
```

*Función que elimina una notificación*

La función “remove\_ToastNotification\_When\_Click” llamará a la función “remove\_ToastNotification” al pulsar sobre la notificación.

```
1  remove_ToastNotification_When_Click(){
2      this.notification_tab.addEventListener('click', ()=>{
3          this.remove_ToastNotification();
4      });
5  }
```

*Función para eliminar una notificación al pulsar sobre ella.*

## Generar una nueva notificación con una función

Adicionalmente, existe una función de nombre “newToastMessage” que recibe un objeto de configuración, esta función creara un nuevo objeto de la clase “createToastMessage” (crea una nueva notificación).

```
1 // create new toast message
2 function newToastMessage(config){
3     let newMessage = new createToastMessage(config);
4 }
```

*Función para crear una nueva notificación.*













## Paquete de iconos

El paquete de iconos incluye símbolos básicos que podrán ser incluidos en conjuntos o separados según los requisitos de su proyecto. Se incluye una mesa de trabajo de **Adobe Illustrator 2020** de nombre “mesa.ai” para poder editar colores, formas, agregar, eliminar o reemplazar iconos y poder ajustarlos según requiera.













**Importante:** Originalmente siguen una dimensión de *40px x 40px* con una separación de *20px* entre cada uno.

## Capas

En la mesa de trabajo encontraras algunas capas con nombres como: “icono a1”, “icono b1”, etc. Esto representa al icono según su posición en la tabla.

		X			
		1 (A)	2 (B)	3 (C)	4 (D)
Y	1				
	2				
	3				

**Tabla de clases CSS por icono**

Icono	Clase CSS
	notification-tab__icon--alert
	notification-tab__icon--information
	notification-tab__icon--cross
	notification-tab__icon--check
	notification-tab__icon--cloud
	notification-tab__icon--internet
	notification-tab__icon--no-internet
	notification-tab__icon--download
	notification-tab__icon--message
	notification-tab__icon--bell
	notification-tab__icon--pin
	notification-tab__icon--reload

## Nombre de clases cuando se utiliza el sprite en “toast notification”

Cuando se haga uso de los iconos en su forma de “sprite” para generar una notificación, el icono se encontrará en una etiqueta “div” con la clase: “notification-tab\_\_icon”.

```
1 <div class="notification-tab__icon">
2 </div>
```

*Elemento HTML que representa un icono.*

Las propiedades que tendrá el elemento al aplicar esta clase serán:

- Ancho: *2.188rem* (Comúnmente *35px*).
- Alto: *2.188rem* (Comúnmente *35px*).
- Desbordamiento: Ocultar.
- URL del fondo: (Asignar según la ubicación de la imagen).
- Posición del fondo: centrar
- Tamaño del fondo: *12.5rem x 9.625rem* (Comúnmente *200px x 154px*).
- Repetir el fondo: No repetir

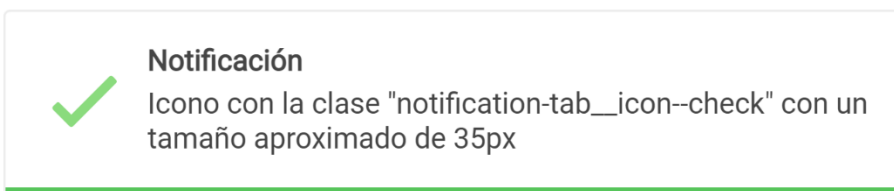
Para indicar el icono a usar, es necesario agregar otra clase al elemento, esta dependerá del icono que desea usar y el estilo que aplicará será:

- Posición del fondo: *(X)rem (Y)rem*.

```
1 <div class="notification-tab__icon notification-tab__icon--alert">
2 </div>
```

*Icono con la clase “notification-tab\_\_icon—alert”*

Al hacer esto, el icono deberá mostrarse correctamente dentro de la ficha de notificación



*Ejemplo de notificación generada con icono según su clase.*

**Nota:** El autor no será responsable por ningún daño o pérdida resultante del uso de este código o paquete de iconos.

**Autor:** José Eduardo Espinoza Díaz.

**Nombre del paquete:** Toast Notification y paquete de iconos básicos.

**Fecha:** 15/02/2023.

**Licencia:** Licencia MIT (<https://opensource.org/licenses/MIT>).