

Proyecto #2

Investigación sobre Lazy Loading y API Intersection Observer.

Lazy Loading

¿Que es Lazy Loading?

Lazy Load es un pequeño script creado por Mika Tuupola, que se apoya en la librería jQuery para configurar la forma en que deben cargar las imágenes insertadas en una página web. Solo pesa 4 KB y es muy fácil de implementar en cualquier página.

Es una técnica que aplaza la carga de recursos no esenciales en el tiempo de carga de la página. Estos recursos no esenciales se cargan en el momento en que son necesarios.

Cómo usar Lazy Load para aplazar la carga de imágenes

Para usar Lazy Load necesitamos hacer los siguientes cambios:

1. Agregar una clase de nombre "lazy" a las imágenes necesarias.
2. En vez de usar el clásico atributo src para especificar la ruta a los archivos de imágenes, usar "data-original".

```
<img class="lazy" data-original="ruta a la imagen.jpeg" width="540" height="375">
```

3. Cargar las librerías JQuery y Lazy Load de la siguiente forma.

```
https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js
```

```
https://cdnjs.cloudflare.com/ajax/libs/jquery.lazyload/1.9.1/jquery.lazyload.min.js
```

4. Agregar un script que contiene la instrucción con la forma en que Lazy Load debe cargar las imágenes.

```
<script>
$(function() {
    $("img.lazy").lazyload();
});
</script>
```

Opciones y efectos que permite Lazy Load

1. Threshold. Permite especificar que las imágenes carguen un número determinado de píxeles antes de aparecer en la parte visible de la pantalla, de esta forma nos aseguramos que se muestran completamente.

2. Eventos. Permite hacer que las imágenes carguen solo con un evento, cuando el usuario da un clic o cuando sitúa el ratón sobre ellas.
3. Efectos. Con esta opción el plugin espera que la imagen cargue completamente y a continuación, la muestra usando un efecto.

Donde descargar o cargar Lazy Load y jQuery

Lazy Load se puede descargar gratis desde la [página del proyecto](#) en Github, en la que también podemos encontrar varios ejemplos prácticos para su uso e implementación.

La última versión de jQuery se puede descargar desde su sitio en: <http://jquery.com/download/>

Los dos archivos también se pueden cargar directamente desde CDNJS, el CDN de CloudFlare.

- [Enlace a JQuery en CDNJS](#)
- [Enlace a LazyLoad en CDNJS](#)

Alternativas de Lazy Loading

Unveil

Es otra librería similar disponible en Github, creada por Luis Almeida, la que reduce Lazy Load al máximo, para usar solo sus opciones elementales.

Los que no necesiten usar los efectos que ofrece Lazy Load, pueden usar en sus proyectos Unveil.

Incluye soporte para dispositivos con pantalla Retina, que agradecen imágenes con una mayor resolución.

- Descarga [Unveil en Github](#)
- Enlace a [Unveil en CDNJS](#)

Lazysizes

Es un script que permite posponer o retrasar la carga de imágenes, marcos (iframe), widgets y otros elementos, hasta que aparezcan en el viewport del navegador.

Solo requiere de la carga de la librería JavaScript "lazysizes.min.js" de 7 KB y agregar la clase "lazyload".

¿Por qué cargar de forma diferida las imágenes y los videos en lugar de solo cargarlos?

Para evitar cargar elementos que es probable que el usuario nunca vea. Esto es un problema por diversos motivos:

- Se desperdician datos. En las conexiones sin medición de uso, esto no es lo peor que puede pasar aunque se puede utilizar ese valioso ancho de banda para descargar otros recursos que el usuario efectivamente verá. Sin embargo, en los planes de datos limitados, la carga de elementos que el usuario nunca verá puede ser un desperdicio real de dinero.
- Se desperdician tiempo de procesamiento, batería y otros recursos del sistema. Una vez cargado el recurso de medios, el navegador debe decodificarlo y representar su contenido en la ventana de visualización.

Al cargar de forma diferida imágenes y videos, se reducen el tiempo de carga inicial de la página, el peso inicial de la página y el uso de recursos del sistema, lo que afecta positivamente el rendimiento.

API Intersection Observer

¿Qué es la API Intersection Observer?

Permite detectar la visibilidad de un elemento, si está en la ventana gráfica actual, y también la visibilidad relativa de dos elementos en relación entre sí.

Hace que sea realmente sencillo detectar cuándo un elemento ingresa a la ventana gráfica y realizar una acción cuando lo hace. Es una API fácil de usar y leer.

El núcleo de la API de Intersection Observer son las siguientes dos líneas.

```
const observer = new IntersectionObserver (callback,  
opciones);  
observer.observe (objetivo);
```

Nota: Intersection Observer no es compatible con todos los navegadores.

Bibliotecas

- [Lazysizes](#) es una biblioteca de GitHub de carga diferida con todas las funciones para cargar de forma diferida imágenes e iframes. Enlaza automáticamente a una clase lazyload en los elementos y requiere especificar URL de imagen en los atributos data-src y/o data-srcset, cuyo contenido se intercambia por los atributos src y/o srcset, respectivamente. Utiliza Intersection Observer (donde se puede aplicar polyfill) y se puede ampliar con [diversos complementos](#) para tareas como la carga diferida de videos.
- [Lozad.js](#) es una opción súper ligera que solo utiliza Intersection Observer. Por esto, es altamente eficaz, pero requiere polyfill para poder utilizarla en navegadores anteriores.
- [Blazy](#) es otra opción similar que se factura como un cargador diferido ligero (su tamaño es 1.4 KB). Al igual que con los tamaños diferidos, no requiere utilidades de terceros para la carga y funciona con IE7+. Lamentablemente, no utiliza Intersection Observer.
- [Yall.js](#) es una biblioteca de mi autoría que utiliza IntersectionObserver y se revierte a los controladores de eventos. Es compatible con IE11 y los principales navegadores.
- Si buscas una biblioteca de carga diferida específica para React, puedes considerar el uso de [react-lazyload](#). Si bien no utiliza Intersection Observer, sí proporciona un método conocido de carga diferida de imágenes para los usuarios habituados a desarrollar aplicaciones con React.