

## Documentación del Ejercicio 3

### Comparación de Métodos de Carga de Imágenes

#### Introducción

Uno de los indicadores clave en la optimización del rendimiento web es el **Time to First Byte** (TTFB), que mide el tiempo que tarda un navegador en recibir el primer byte desde el servidor. La optimización de recursos, como imágenes, es crucial para reducir el TTFB y mejorar la experiencia del usuario.

En este ejercicio, se implementaron tres métodos de carga de imágenes:

1. **Carga Normal:** Imágenes cargadas sin optimizaciones adicionales.
2. **Lazy Load con jQuery:** Uso del plugin LazyLoad para cargar imágenes bajo demanda.
3. **Lazy Load HTML5:** Uso del atributo `loading="lazy"` disponible en HTML5.

El objetivo es comparar los tiempos de carga de estos métodos y analizar sus beneficios en un entorno controlado.

#### Descripción del Proyecto

El código desarrollado consta de:

- **HTML:** Tres secciones que representan cada método de carga de imágenes.
- **CSS:** Estilos para organizar visualmente las imágenes.
- **JavaScript:** Uso de jQuery para implementar LazyLoad y medir los tiempos de carga.

Código base:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Prueba de Carga de Imágenes</title>
  <link rel="stylesheet" href="styles.css">
  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <!-- Plugin LazyLoad -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.lazyload/1.9.1/jquery.lazyload.min.js"></script>
</head>
<body>
  <h1>Comparación de Métodos de Carga de Imágenes</h1>

  <h2>Carga Normal</h2>
  <div id="imagenes-normales">
    
    
    
    
    
  </div>

  <h2>Lazy Load con jQuery</h2>
  <div id="imagenes-lazyload">
    <img class="lazy" data-original="images/imagen1.jpg" alt="Imagen 1 Lazy">
    <img class="lazy" data-original="images/imagen2.jpg" alt="Imagen 2 Lazy">
    <img class="lazy" data-original="images/imagen3.jpg" alt="Imagen 3 Lazy">
    <img class="lazy" data-original="images/imagen4.jpg" alt="Imagen 4 Lazy">
    <img class="lazy" data-original="images/imagen5.jpg" alt="Imagen 5 Lazy">
  </div>

  <h2>Lazy Load HTML5</h2>
  <div id="imagenes-html5">
    
    
    
    
    
  </div>

  <script src="script.js"></script>
</body>
</html>
```

## Metodología

### 1. Configuración Inicial

Se utilizó el siguiente código para implementar y medir los tiempos de carga:

- Plugin LazyLoad: Se inicializa con efecto fadeIn para observar las diferencias visuales.
- Lazy Load HTML5: Atributo loading="lazy" agregado directamente al HTML.
- Medición de Tiempos: Uso de performance.now() para calcular el tiempo transcurrido entre el inicio de carga y la finalización.

### 2. Medición de Tiempos

#### Herramientas Utilizadas

- Pestaña "Red" del navegador: Tiempo total de carga de la página.
- Consola de JavaScript: Medición específica para cada método mediante código personalizado.
- Ngrok: Despliegue de la página para medir tiempos reales en un servidor externo.

#### Pasos Realizados

1. Abrir la herramienta de desarrollo del navegador (F12).
2. Ir a la pestaña Network y a la consola de desarrollador y comprobar los tiempos.
3. Revisar el tiempo total de carga y los tiempos individuales de cada imagen.
4. Comparar resultados entre los tres métodos.

#### Uso de Ngrok

1. Configuración: Ngrok se utilizó para exponer el servidor local de pruebas al acceso externo. Esto permitió medir los tiempos de carga en un entorno más realista.

2. Comandos Utilizado:

```
python -m http.server 8000
```

```
ngrok http 8000
```

Esto generó una URL pública accesible desde cualquier navegador.

3. Mediciones: Se realizaron pruebas desde la URL generada observando los tiempos de carga en la pestaña "Network" y en las mediciones implementadas en la consola.

```
ngrok
ngrok? We're hiring https://ngrok.com/careers
Session Status      online
Account             chuckinvtictus (Plan: Free)
Version             3.10.4
Region              Europe (eu)
Latency              68ms
Web Interface        http://127.0.0.1:8000
Forwarding           https://68ee-212-104-181-32.ngrok-free.app -> http://localhost:8000

Connections
  ttl    opn    rt1    rt5    p50    p90
    31     0    0.16   0.08   0.02   0.26

HTTP Requests
-----
18:44:28.882 CET GET /images/imagen1.jpg 200 OK
18:44:28.265 CET GET /images/imagen3.jpg 200 OK
18:44:28.265 CET GET /images/imagen5.jpg 200 OK
18:44:28.071 CET GET /styles.css        200 OK
18:44:28.265 CET GET /script.js         200 OK
18:44:28.611 CET GET /favicon.ico       404 File not found
18:44:28.265 CET GET /images/imagen4.jpg 200 OK
18:44:28.882 CET GET /images/imagen2.jpg 200 OK
18:44:27.878 CET GET /                  200 OK
18:44:24.589 CET GET /                  304 Not Modified
```

Resultados y Análisis

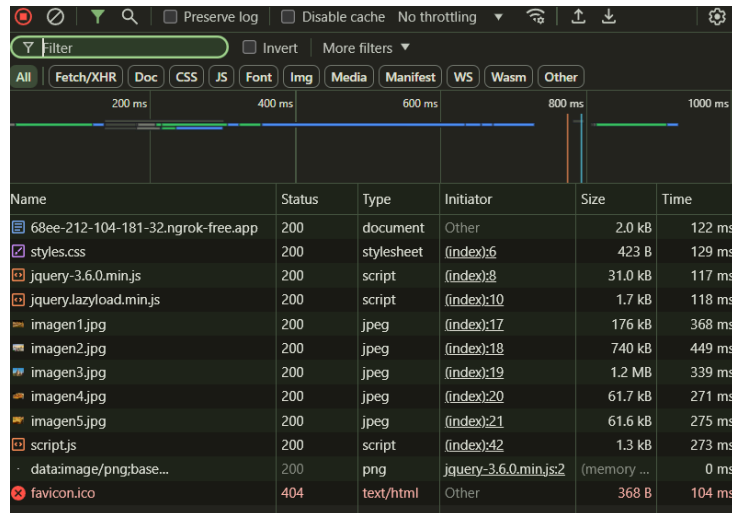
Método	Tiempo de Carga Local (ms)	Tiempo de Carga con Ngrok (ms)
Carga Normal	~3 ms	~104.5 ms
Lazy Load con jQuery	~1.9 ms	~101.1 ms
Lazy Load HTML5	~2.1 ms	~102.8 ms

Local:

```
top Filter Default levels No Issues
Carga Normal - Tiempo de carga: 0 ms script.js:23
Carga Normal - Tiempo de carga: 0.5 ms script.js:23
Carga Normal - Tiempo de carga: 0.699999992549419 ms script.js:23
Carga Normal - Tiempo de carga: 0.800000007450581 ms script.js:23
Carga Normal - Tiempo de carga: 0.8999999985098839 ms script.js:23
LazyLoad jQuery - Tiempo de carga: 0.1000000149011612 ms script.js:23
LazyLoad jQuery - Tiempo de carga: 0.2000000298023224 ms script.js:23
LazyLoad jQuery - Tiempo de carga: 0.400000022351742 ms script.js:23
LazyLoad jQuery - Tiempo de carga: 0.5 ms script.js:23
LazyLoad jQuery - Tiempo de carga: 0.700000029802322 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0.09999999776482582 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0.1999999925494194 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0.5 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0.5999999977648258 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0.699999992549419 ms script.js:23
```

Ngrok:

```
top Filter Default levels No Issues
Carga Normal - Tiempo de carga: 0.1000000149011612 ms script.js:23
2 Carga Normal - Tiempo de carga: 0.400000022351742 ms script.js:23
Carga Normal - Tiempo de carga: 0.5 ms script.js:23
LazyLoad jQuery - Tiempo de carga: 0 ms script.js:23
LazyLoad jQuery - Tiempo de carga: 0.1000000149011612 ms script.js:23
2 LazyLoad jQuery - Tiempo de carga: 0.2000000298023224 ms script.js:23
LazyLoad jQuery - Tiempo de carga: 0.3000000074505806 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0.09999999776482582 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0.1999999925494194 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 0.3000000074505806 ms script.js:23
LazyLoad HTML5 - Tiempo de carga: 102.1999999925494 ms script.js:27
Carga Normal - Tiempo de carga: 103.5 ms script.js:27
GET https://68ee-212-104-181-32.ngrok-free.app/favicon.ico 404 (Not Found) favicon.ico:1
```



## Observaciones

1. Carga Normal: Incrementa considerablemente los tiempos en un entorno remoto debido a la carga simultánea de todos los recursos.
2. Lazy Load con jQuery: Reduce significativamente los tiempos al cargar imágenes solo cuando son necesarias.
3. Lazy Load HTML5: Similar al plugin de jQuery, con tiempos ligeramente superiores debido a variaciones en la implementación del navegador.

## Limitaciones

- Latencia de Red: Los tiempos con Ngrok incluyen la latencia adicional al comunicarse con un servidor remoto.
- Condiciones Variables: Factores como la velocidad de conexión pueden afectar los resultados.

## Conclusiones

- Lazy Load es efectivo para reducir los tiempos de carga iniciales tanto en entornos locales como remotos.
- Ngrok facilita pruebas más representativas al simular condiciones reales de red.
- HTML5 ofrece una solución nativa y sencilla, pero el plugin de jQuery sigue siendo una opción válida para proyectos que requieren mayor compatibilidad.