

# RENDIMIENTO EN FRONTEND



`luisddm.github.io/web-performance-salamancadevs`

**LUIS DE DIOS**

# **PRESENTACIÓN**

# QUIÉN SOY

- Ingeniero de Teleco
- Últimos cuatro años especializado en frontend
- Ahora dedicado a backend y sistemas en Alea Soluciones
- Coorganizador de FrontFest
- Aficionado a la seguridad y la electrónica

# **DISCLAIMER**

# MOTIVACIÓN

# **TENEMOS UN PROBLEMA CON LA WEB**

Tamaño, lentitud, complejidad → falta de usabilidad

No se pensó para su uso actual

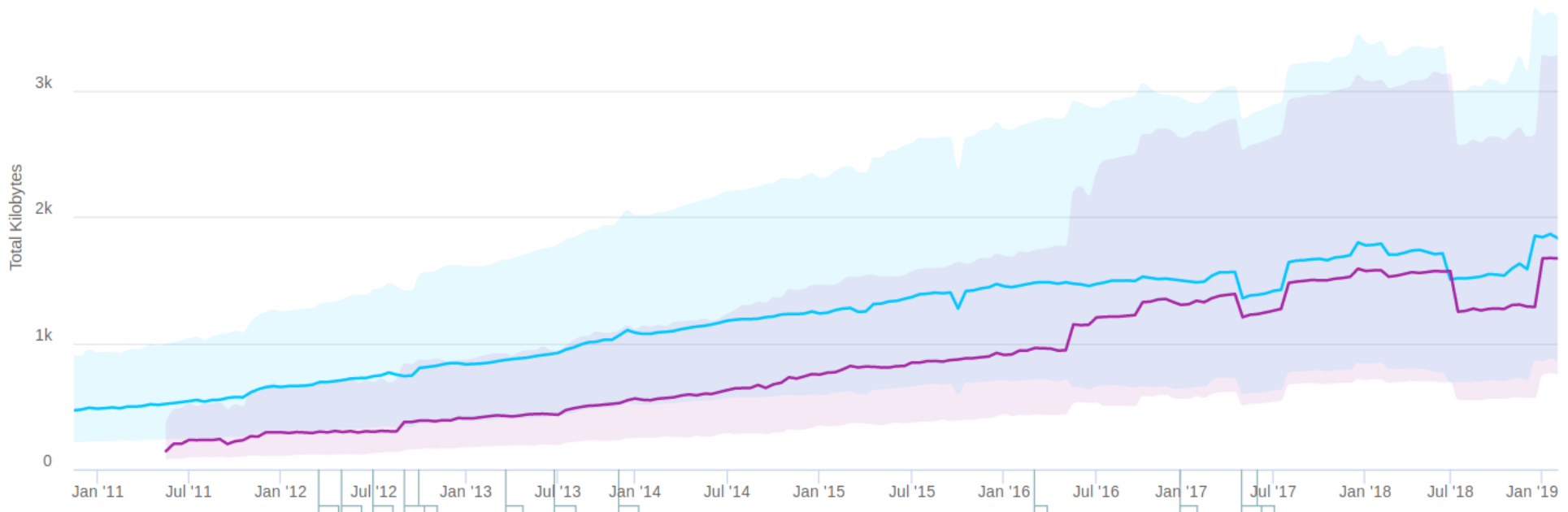
# ANÁLISIS

# CÓMO HEMOS CAMBIADO

- De simple texto con hipervínculos a grandes aplicaciones interactivas con imágenes, videos, etc.
- De compartir conocimiento a negocio
- De unos pocos "frikis" al público en general



# EVOLUCIÓN DEL PESO DE LA WEB



→ HTTP Archive: State of the Web

# **CASO DE ESTUDIO: USA TODAY**

Welcome to USA TODAY NETWORK'S EUROPEAN UNION  
EXPERIENCE

[Learn More](#)



[SUBSCRIBE NOW](#)  
to get home delivery

NEWS SPORTS LIFE MONEY TECH TRAVEL OPINION 61° CROSSWORDS INVESTIGATIONS MORE [Subscribe](#) [Sign In](#)



## Supreme Court: Roberts, Kavanaugh stall conservative takeover

10:51 AM EDT Apr 7, 2019

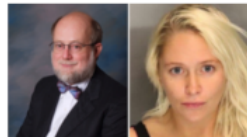
### Dickie V: 'Absurd' missed call decided Auburn game

9:38 AM EDT Apr 7, 2019



### 2 more face charges in psychiatrist's slaying

10:18 PM EDT Apr 6, 2019



### A presidential secret all investors need to know



WASHINGTON

## Supreme Court: Roberts, Kavanaugh stall conservative takeover

[Read Story](#) [Supreme Court's new conservative majority](#)  
[Committee OKs controversial Kavanaugh replacement](#)



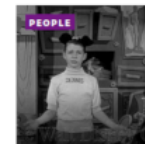
Izzo, Spartans flop in the  
Final Four yet again



'SNL' mocks Joe Biden's  
apology video



Boeing 737 Max: Will  
skittish passengers  
return?



Body found at home of  
missing Disney  
Mouseketeer



#### TOP STORIES

- [Dickie V: 'Absurd' missed call decided Auburn game](#)
- [2 more face charges in psychiatrist's slaying](#)
- [A presidential secret all investors need to know](#)
- [American Airlines extends Max 8 flight cancellatio...](#)
- [Opinion: America's 'boy crisis' more than economic](#)
- [CNN's Don Lemon engaged to boyfriend Tim Malone](#)
- [Watch Charles Barkley's roller coaster of emotions...](#)
- [Your kids could be key to a bigger tax break](#)
- [Who has edge in Texas Tech-Virginia title game?](#)

[Share your feedback](#) to help improve  
our site experience!

#### AD CONTENT



Yahoo Search  
The Early Signs of Type 2 Diabetes.  
Learn About Treatments & Symptoms



Teeth Whitening By Snow  
Want The Real Answer To Celebrity

# RESULTADOS

- **Edición Europa:** 34 requests / 540 kB
- **Edición USA:** ~400 requests / ~10 MB

→ USA Today

→ Twit de @paulcalvano

# OTROS CASOS DE ESTUDIO

Sitio web	Tx / Total	Req.
Marca	5,6 / 13,8 MB	~600
El País	6,1 / 17,6 MB	~450
Tumblr	3,6 / 6,2 MB	~75
Medium	1,4 / 3,3 MB	~50

**¿A QUIÉN PERJUDICA ESTO?**

# **SESGO DEL DESARROLLADOR (Y DE NEGOCIO)**

La mayoría de los que nos dedicamos a esto tenemos buen hardware y buenas conexiones, pero el "mundo real" es diferente

# ÁMBITO GEOGRÁFICO

Algunas de las webs que hacemos las utiliza gente que vive en entornos muy diferentes

→ Mapa cobertura Movistar 4G

→ Muchas zonas de Europa aún carecen de conexión de calidad a Internet (El País)



# DISPOSITIVOS HETEROGÉNEOS

Existe una enorme variedad de dispositivos para acceder a la web, que cuestan desde pocas decenas a miles de euros

# CONSUMO DE DATOS

¿Cuántas webs podemos ver con un giga de datos?

# **BAJO "ENGAGEMENT"**

Cuando una web tarda demasiados segundos en cargar, la abandonamos

# **ACCESIBILIDAD**

El mal rendimiento implica exclusión de algunos  
segmentos de población

# PRIVACIDAD

Enorme cantidad de trackers que envían datos personales permanentemente a grandes empresas de internet, incluso sin estar logueados

También impacta en el rendimiento, puede haber scripts bloqueantes

**CONCEPTOS**

# PROGRESSIVE ENHANCEMENT

- Implica una forma integral de pensar y desarrollar, afecta a todo el pipeline
- Primero el contenido fundamental, y las mejoras se activan en función de las capacidades del hardware
- Que TODOS los usuarios tengan una experiencia mínima, que no sea todo o nada

→ [The cost of Javascript \(Addy Osmani\)](#)





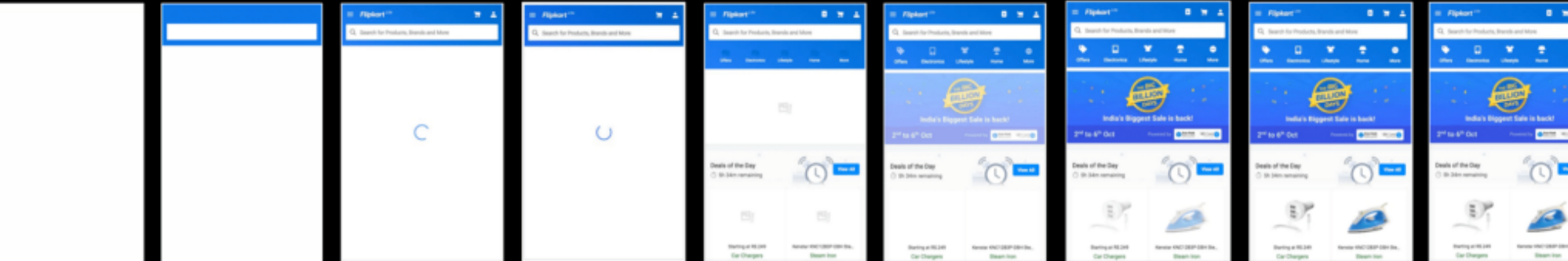
# **FIRST MEANINGFUL PAINT**

Que nos "aporte algo" lo más rápidamente posible

is it happening?

is it useful?

is it usable?



Navigation begins

First Contentful Paint

First Meaningful Paint

Visually ready

Fully Loaded

Time to first byte

Navigation has successfully started

Page's primary content is visible

Page looks nearly done

End of load lifecycle

First Paint

Time to Interactive

The first non-blank paint on screen

Visually usable and engaging

# PRINCIPIO DE PARETO

También conocido como regla del 80/20

Disponemos de tiempo y presupuestos limitados, así que hay que priorizar las optimizaciones

**SOLUCIONES**

# IMÁGENES

- En cuanto a tamaño, son lo más costoso
- Identificar cuándo usar JPG, PNG, WebP, SVG...
- Balance adecuado entre calidad, tamaño y dimensiones

# CARGAR SÓLO IMÁGENES NECESARIAS

Lazy loading → scroll vertical o sliders

Usar diferentes tamaños o resoluciones de imágenes  
para diferentes media queries

→ [How to Build Responsive Images with srcset](#)  
(Sitepoint)

# CSS VS HTML

```
.image {  
  background-image: url(pic.jpg)  
}  
@media (min-width: 800px) {  
  .image {  
    background-image: url(pic-2x.jpg)  
  }  
}
```

```
<picture>  
  <source srcset="pic-2x.jpg" media="(min-width: 800px)">  
    
</picture>
```

# MANIPULACIÓN DE IMÁGENES

- ImageOptim, XNConvert, The GIMP, imagemagick, svgo...
- Automatizar (Gulp/gulp-imagemin, Grunt, Webpack o simplemente scripts)
- Eliminar metadatos
- Guardar con entrelazado



# AUTOMATIZACIÓN

## Ejemplo con Gulp

```
gulp.task('images', function() {  
  return gulp.src('pic.jpg')  
    .pipe(imageResize({ width: 800 }))  
    .pipe(imagemin({ progressive: true }))  
    .pipe(gulp.dest('./dist/img'))  
    .pipe(rename({ suffix: '@2x' }))  
    .pipe(imageResize({ width: 400 }))  
    .pipe(imagemin({ progressive: true }))  
    .pipe(gulp.dest('./dist/img'))  
});
```

# LO MISMO CON LOS VÍDEOS

- Elegir el formato adecuado (WebM, MP4, OGG...) pero cuidado con la compatibilidad
- ¡No GIFs animados! Usar vídeo en su lugar
- ¡No autoplay! Que decida el usuario
- Cuidado con incrustar vídeos de YouTube

→ [El Elemento Video en MDN](#)

# CONVERSIÓN DE VÍDEOS

Ejemplo: de GIF a MP4

```
$ ffmpeg -i anim.gif -b:v 0 -crf 40 -vf scale=600:-1 video.mp4
```

## Salamanca Devs

@DevsSalamanca

Somos un grupo developers apasionados por la tecnología y el aprendizaje. Organizamos charas, meetups y eventos en Salamanca. ¡Únete!

📍 Salamanca, Spain

📅 Se unió en enero de 2019

📷 Fotos y videos



Tweets

Tweets y respuestas

Multimedia



**Salamanca Devs** @DevsSalamanca · 7 abr.

Apuntarse en el meetup!! 📍 [meetup.com/es-ES/Salamanc...](https://meetup.com/es-ES/Salamanc...)



1



2

🔍 Consola 📄 Depurador {} Editor de estilos 🔄 Rendimiento 📦 Memoria ⬆️ Red 🗄️ Almacenamiento 🧑 Accesibilidad

🔍 Buscar en HTML

```
<div style= cursor: pointer; height: 100%; width: 100%; position: relative , arial; line-height:
normal; transform: translateZ(0px);"> [event]
  <div role="button" style="position: absolute; height: 100%; width: 100%;" tabindex="0"> [event]
    <div style="position: relative; width: 100%; height: 100%; background-color: transparent;
overflow: hidden;">
      <video preload="auto" playsinline="" style="width: 100%; height: 100%; position: absolute;
background-co...ck; top: 0%; left: 0%; transform: rotate(0deg) scale(1.005);"
poster="https://pbs.twimg.com/tweet_video_thumb/D3igUqsX4AAesJd.jpg"
src="https://video.twimg.com/tweet_video/D3igUqsX4AAesJd.mp4" type="video/mp4"></video> [event]
    </div>
```



# MINIFICAR

- Eliminar espacios, intros, reducir nombres de variables, etc. (UglifyJS)
- El tooling propio de cada framework suele automatizar esto para diferentes entornos
- También se pueden usar CDNs para código de terceros

# SOURCE MAPS

- Es un mapeo del código minificado para poder reconstruirlo en el navegador y así poder debuggear
- Se enlaza desde el fichero minificado y se carga únicamente cuando accedemos al código con las Dev Tools

```
//# sourceMappingURL=main.js.map
```

# CONCATENAR

- Juntar todos los archivos JS y CSS en uno solo para reducir las peticiones HTTP
- Pero a veces nos interesa más dividirlos e irlos cargando según se necesiten al navegar
- Si estamos en un framework podemos usar dynamic imports → Lazy-load router



# DYNAMIC IMPORT

Ejemplo: router de Vue.js

```
export default new Router({  
  routes: [  
    {  
      path: '/',  
      name: 'home',  
      component: Home  
    },  
    {  
      path: '/about',  
      name: 'about',  
      component: () => import('./views/About.vue')  
    }  
  ]  
})
```

# ELIMINAR CÓDIGO MUERTO

- Tree-shaking con los módulos de Javascript (Webpack 4)
  - Comprobar con tests unitarios, funcionales y E2E
- Eliminar CSS sin usar
  - Comprobar con tests visuales (ej Chromatic)

# TIPOGRAFÍAS

- Usar formatos más ligeros (woff2), pero sin romper compatibilidad
- Usar `font-display: swap` para que el texto sea visible con una fuente de fallback mientras se carga la nuestra
- Usar SVGs en lugar de fuentes de iconos

→ La regla `@font-face` de CSS

# LA REGLA @FONT-FACE

```
@font-face {  
  font-family: "Open Sans";  
  src: url("/fonts/OpenSans-Regular.woff2") format("woff2"),  
        url("/fonts/OpenSans-Regular.ttf") format("ttf");  
  font-weight: 400;  
  font-display: swap;  
}
```

# CACHEADO EFICIENTE

- Usar HTTP ETag para establecer adecuadamente los tiempos de vida
- Un ETag es un checksum para comprobar si un recurso ha cambiado
- Con las cabeceras Cache-Control y Expire podemos establecer el TTL

# COMPRESIÓN

El navegador y el servidor negocian la compresión

- Request

```
GET /route HTTP/1.1  
HOST example.com  
Accept-Encoding: gzip, deflate
```

- Response

```
HTTP/1.1 200 OK  
Content-Encoding: gzip
```

# APIS EXPERIMENTALES

Ejemplo, sólo funciona en Chrome

```
networkInformation = navigator.connection // '4g'
```

Útil sobre todo en móviles

→ [navigator.connection](#) en MDN

# OTROS DETALLES

- Normalmente colocar scripts al final del `<body>` para no bloquear el renderizado HTML y CSS
- Evitar redirecciones
- Evitar demasiados nodos HTML
- Marcar scripts no críticos con `defer`



# HTTP/2

- Ha de estar soportado tanto por el servidor como por el navegador
- Modifica algunas técnicas vistas previamente

→ [Introducción a HTTP/2](#)

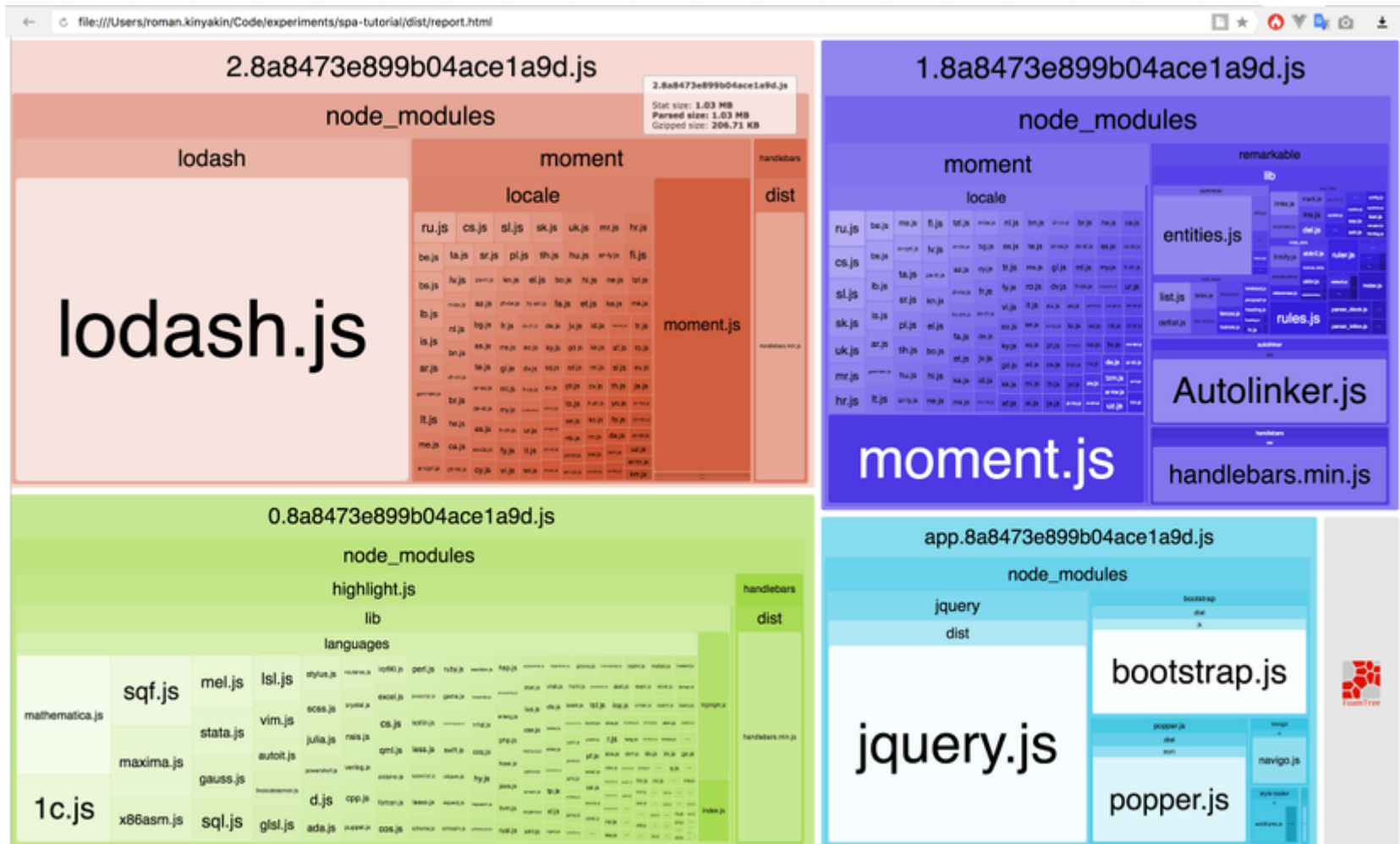
# **HERRAMIENTAS DE ANÁLISIS**

# IMPORT-COST

Extensión para VSCode que muestra el tamaño al lado de cada import o require

```
const {uniqueId} = require('lodash') // 70KB  
const uniqueId = require('lodash/uniqueId') // 2KB
```

# WEBPACK BUNDLE ANALYSER



# BUNDLEPHOBIA

lodash@4.17.11



Lodash modular utilities

no dependencies



## BUNDLE SIZE

**69.1 kB**

MINIFIED

**24.2 kB**

MINIFIED + GZIPPED

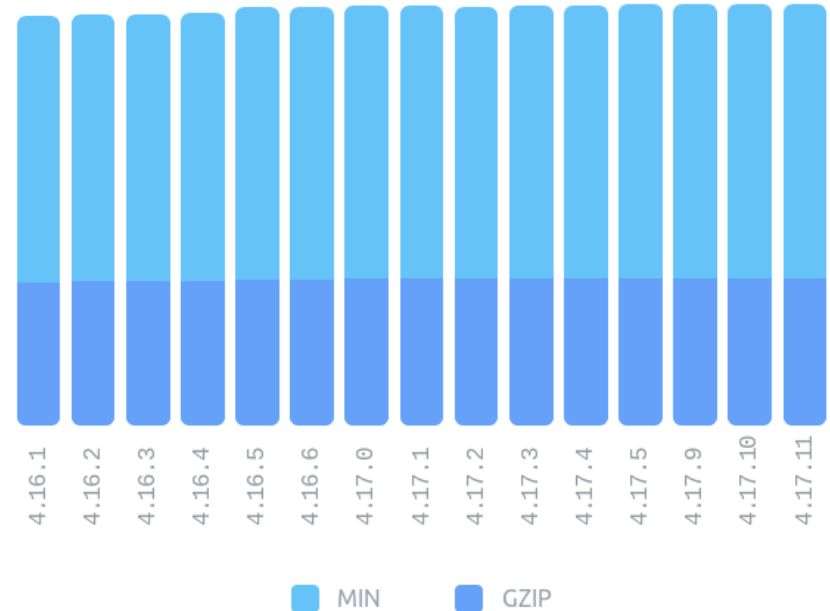
## DOWNLOAD TIME

**0.81 s**

2G EDGE ⓘ

**485 ms**

EMERGING 3G ⓘ



# LIGHTHOUSE



Performance



Accessibility



Best Practices



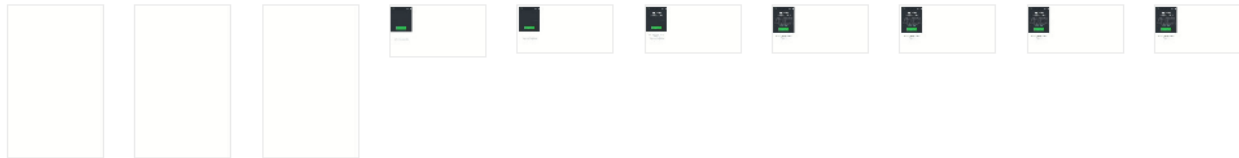
SEO



Progressive Web App

First Contentful Paint	2.7 s ⓘ	First Meaningful Paint	4.2 s ⚠
Speed Index	4.9 s ⓘ	First CPU Idle	5.7 s ⓘ
Time to Interactive	13.4 s ⚠	Estimated Input Latency	100 ms ⓘ

Values are estimated and may vary.



## Opportunities

These optimizations can speed up your page load.

Opportunity		Estimated Savings
1	Defer offscreen images	<div></div> 14.58 s ▾
2	Properly size images	<div></div> 9.32 s ▾
3	Serve images in next-gen formats	<div></div> 8.89 s ▾
4	Eliminate render-blocking resources	<div></div> 1.57 s ▾
5	Efficiently encode images	<div></div> 1.2 s ▾

# GTMETRIX

## Performance Scores



## Page Details



### PageSpeed

### YSlow

### Waterfall

### Timings

### Video

### History

RECOMMENDATION	GRADE		TYPE	PRIORITY
▼ <a href="#">Serve scaled images</a>	F (0)	▼	IMAGES	HIGH
▼ <a href="#">Leverage browser caching</a>	F (0)	▼	SERVER	HIGH
▼ <a href="#">Enable gzip compression</a>	D (66)	▼	SERVER	HIGH
▼ <a href="#">Optimize images</a>	D (68)	◆	IMAGES	HIGH
▼ <a href="#">Defer parsing of JavaScript</a>	C (78)	▲	JS	HIGH
▼ <a href="#">Specify image dimensions</a>	A (95)	◆	IMAGES	MEDIUM
▼ <a href="#">Minify HTML</a>	A (99)	◆	CONTENT	LOW
▼ <a href="#">Minify CSS</a>	A (99)	◆	CSS	HIGH
▼ <a href="#">Minify JavaScript</a>	A (99)	▲	JS	HIGH
▼ <a href="#">Specify a Vary: Accept-Encoding header</a>	A (90)	◆	SERVER	LOW
▼ <a href="#">Avoid bad requests</a>	A (100)	◆	CONTENT	HIGH
▼ <a href="#">Avoid landing page redirects</a>	A (100)	◆	SERVER	HIGH

## What do my scores mean?

### Rules are sorted in order of impact upon score

Optimizing rules at the top of the list can greatly improve your overall score.

### Not every recommendation will apply to your page

The recommendations are meant to be generic, best practices; some things will be out of your control (eg. external resources) or may not apply to your page.

[Learn more about PageSpeed/YSlow scores and how they affect performance.](#)

## Need optimization help?

Read our [how to guides](#) and [optimization explained articles](#) for additional direction on improving your page performance.

Meet an expert to help you with

# ENLACES

- Import Cost en Visual Studio Marketplace
- Webpack Bundle Analysis
- BundlePhobia
- Lighthouse
- GTmetrix



# CONCLUSIONES

# CONCLUSIONES

Elaborar un presupuesto realista de assets y cumplirlo

La petición más rápida es la que no se hace

El byte "menos pesado" es el que no se transmite

¡Minimizar el resto!

Establecer prioridades y buscar la simplicidad

# RECURSOS

→ [@PerfReviews\\_](#) en Twitter

# ¡GRACIAS!

## ¿ALGUNA PREGUNTA?

- **Twitter:** @luisddm\_
- **Mastodon:**  
mastodon.technology/@luisddm
- **Medium:** medium.com/@luisddm
- **Github:** github.com/luisddm