



MÓDULO DIW · DAW — UNIDAD DIDÁCTICA 6

Multimedia en la web: formatos, licencias y optimización

RA3 · Integra elementos multimedia analizando su interactividad y usabilidad

- Usar imágenes, audio y vídeo con la licencia adecuada
- Elegir el formato correcto y servir solo el tamaño que hace falta
- Integrar SVG, audio, vídeo y animaciones sin castigar el rendimiento

foto.png

2,4 MB



foto.jpg

310 KB



foto.avif

95 KB



la misma foto, a simple vista igual



DÓNDE ESTAMOS

AURIA ya está bien hecha: ahora, que pese poco



La pregunta de la unidad: ¿cómo ponemos imágenes, audio y vídeo que podamos usar legalmente y que no hagan lenta la web?

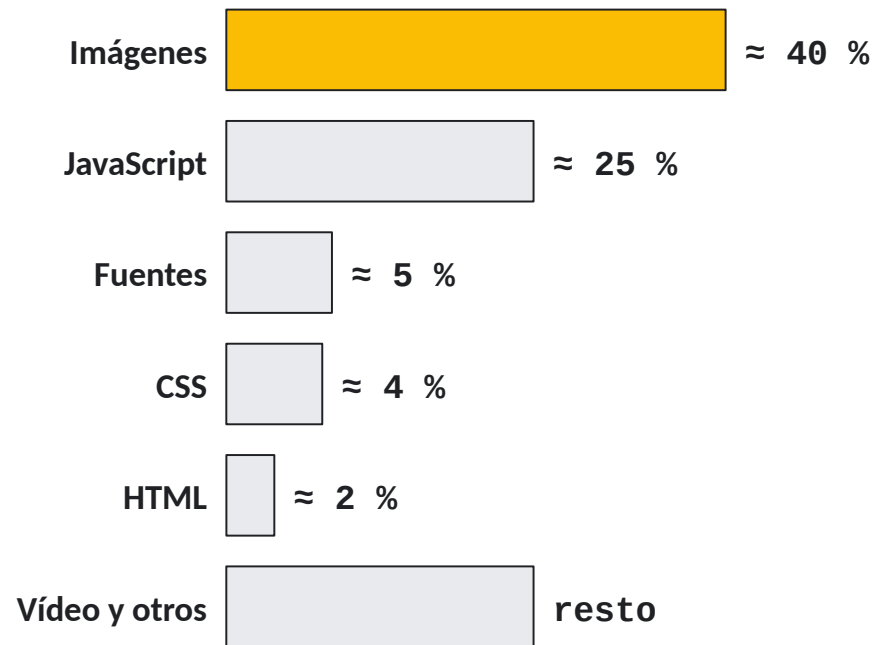
Empieza el RA3: multimedia. Es la unidad más larga del trimestre (12 h).



Las imágenes son lo que más pesa de una web

¿Qué se descarga en una página típica?

reparto orientativo del peso (HTTP Archive)



Velocidad

Cada segundo de espera hace que más gente abandone la página



Datos móviles

Un móvil con tarifa limitada paga cada megabyte que descarga



Derechos de autor

Una foto «cogida de Google» puede acabar en una reclamación



Accesibilidad

Texto alternativo, subtítulos y nada que se mueva sin permiso



Posicionamiento

Google mide la carga (Core Web Vitals) para ordenar resultados



DATO CLAVE — Optimizar imágenes es casi siempre la mejora de rendimiento más barata: no hay que tocar el código de la web.



Seis bloques, de la licencia a la animación



1 • Licencias

¿Puedo usarlo?

RA3 . a



2 • Formatos

JPEG, PNG, WebP, AVIF,
SVG

RA3 . b



3 • Imágenes en HTML

picture, srcset, lazy, CLS

RA3 . b



4 • SVG y tratamiento

Iconos y herramientas

RA3 . c - d



5 • Audio y vídeo

Códecs, etiquetas,
FFmpeg

RA3 . e - g



6 • Animación

Y medir el peso

RA3 . f



Laboratorio: las imágenes, el icono, el vídeo y el audio de AURIA, con licencia comprobada, en el formato y tamaño justos, y midiendo el peso antes y después.



CLAVE — Cada recurso multimedia responde a tres preguntas: ¿puedo usarlo?, ¿en qué formato?, ¿cuánto pesa?



BLOQUE 1

¿Puedo usar esta imagen?

Derechos de autor y licencias del material multimedia (RA3.a)



Por defecto, todo tiene dueño



El derecho de autor nace con la obra: no hace falta registrarla ni poner ©. Si no dice nada, todos los derechos están reservados.

Encontrar no es poder usar: que una imagen aparezca en un buscador no te da ningún permiso.

Lo que manda es la licencia: el permiso que da el autor, y sus condiciones.

En España

La Ley de Propiedad Intelectual protege fotos, ilustraciones, música, vídeos, tipografías y código.

Usar sin permiso una foto en la web de una empresa puede acabar en una reclamación económica.



OJO — AURIA es una tienda: todo lo que pongamos en su web es uso comercial.



Creative Commons: cuatro piezas que se combinan

BY

Atribución

Hay que citar al autor. Está en TODAS las CC.

SA

Compartir igual

Lo que hagas a partir de la obra, con la misma licencia

NC

No comercial

Prohibido usarla para ganar dinero

ND

Sin derivadas

Se usa tal cual: no se recorta, ni se edita, ni se retoca

Ejemplos de combinaciones:

CC BY

la más abierta: usa, modifica y vende, citando

CC BY-SA

la de Wikipedia

CC BY-NC

nada comercial: no vale para AURIA

CC BY-NC-ND

la más cerrada



Y APARTE — CC0 no es una combinación: es renunciar a todos los derechos (dominio público). Sin condiciones.



Qué licencia tiene lo que vas a usar

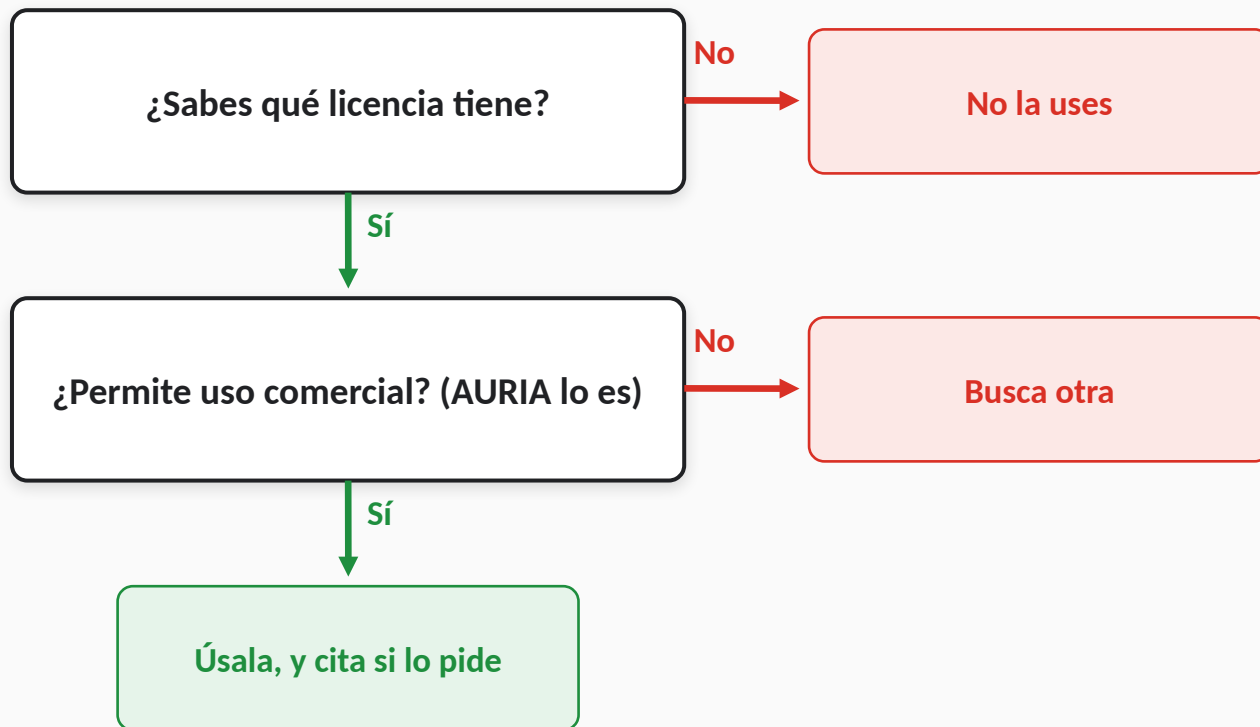
Licencia	Uso comercial	Modificar	Citar	Dónde la encontrarás
© Todos los derechos	Solo con permiso	No	—	Casi todo lo que hay en internet, bancos de pago
CC BY	Sí	Sí	Sí	Wikimedia Commons, Flickr (filtrando)
CC BY-SA	Sí	Sí, misma licencia	Sí	Wikipedia
CC BY-NC	No	Sí	Sí	Material educativo
CC0	Sí	Sí	No	Openverse, museos con colecciones abiertas
Licencia propia del banco	Sí (leer límites)	Sí	No obligatorio	Unsplash, Pexels, Pixabay
SIL OFL (tipografías)	Sí	Sí	—	Google Fonts



OJO — Unsplash o Pexels no son CC0: tienen su propia licencia, muy abierta, pero con límites (por ejemplo, no revender la foto tal cual).



Decidir si se puede usar, y cómo citar



Cómo citar: TALF

- T** Título de la obra (si lo tiene)
- A** Autor con enlace a su perfil
- L** Licencia con enlace a su texto
- F** Fuente dónde la encontraste

«Auriculares» · Ana Ruiz · CC BY 4.0 · Wikimedia Commons



HÁBITO — Apunta la licencia y la fuente en el momento de descargar. Una semana después nadie recuerda de dónde salió.



BLOQUE 2

Formatos de imagen

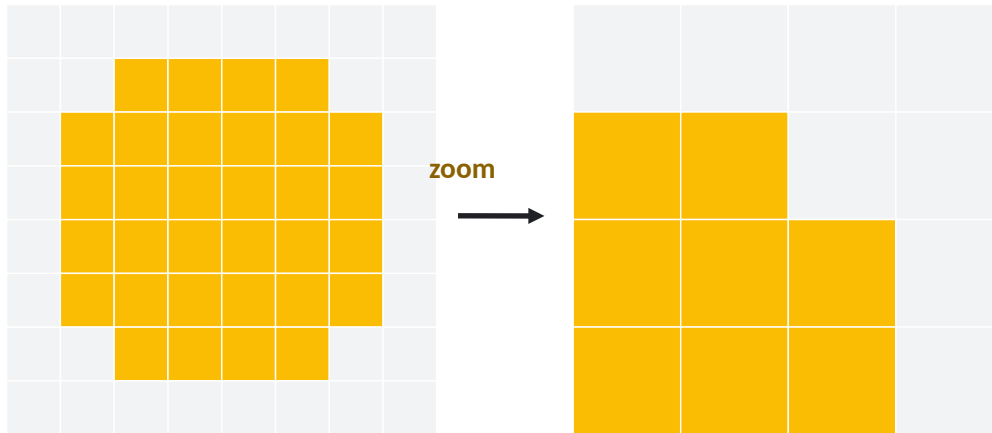
Raster o vector, con o sin pérdida, y cuál elegir en cada caso (RA3.b)



Raster o vectorial: píxeles o instrucciones

Raster

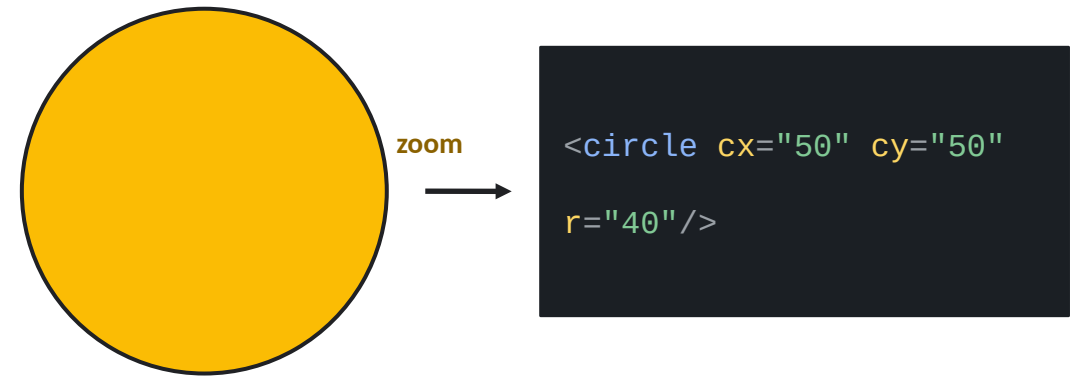
JPEG · PNG · WebP · AVIF · GIF



- Una rejilla de puntos de color
- Al ampliar, se ven los cuadros
- Fotos y todo lo «realista»

Vectorial

SVG



- Formas descritas con números
- Nítido a cualquier tamaño
- Iconos, logos, ilustraciones planas



CLAVE — Un logo en SVG pesa lo mismo a 16 px que a 1600 px. Una foto no se puede pasar a vector: no son formas, son millones de puntos.



Comprimir con pérdida o sin pérdida

Sin pérdida (lossless)

Al descomprimir sale exactamente la misma imagen, píxel a píxel.

PNG · WebP sin pérdida · GIF

Para: Capturas de pantalla, texto, bordes nítidos

Con pérdida (lossy)

Tira detalles que el ojo casi no ve. Cuanto más comprimes, más pesa menos... y peor se ve.

JPEG · WebP · AVIF

Para: Fotografías

El control de calidad (JPEG, WebP y AVIF):

100  pesa mucho

75-80  punto dulce para fotos

30  se ven bloques



OJO — Cada vez que reabres y guardas un JPEG pierde un poco más. Edita siempre desde el original.



Los formatos de imagen, cara a cara

Formato	Compresión	Transparencia	Animación	Soporte	Úsalo para
JPEG	Con pérdida	No	No	Universal	Fotos (formato de respaldo)
PNG	Sin pérdida	Sí	No	Universal	Capturas, gráficos con transparencia
GIF	Sin pérdida, 256 colores	Sí / no	Sí	Universal	Casi nada hoy
WebP	Las dos	Sí	Sí	Todos los actuales	Sustituto general de JPEG y PNG
AVIF	Las dos	Sí	Sí	Todos los actuales	Fotos: el que menos pesa
SVG	Es texto (vector)	Sí	Con CSS/JS	Universal	Iconos, logos, ilustraciones

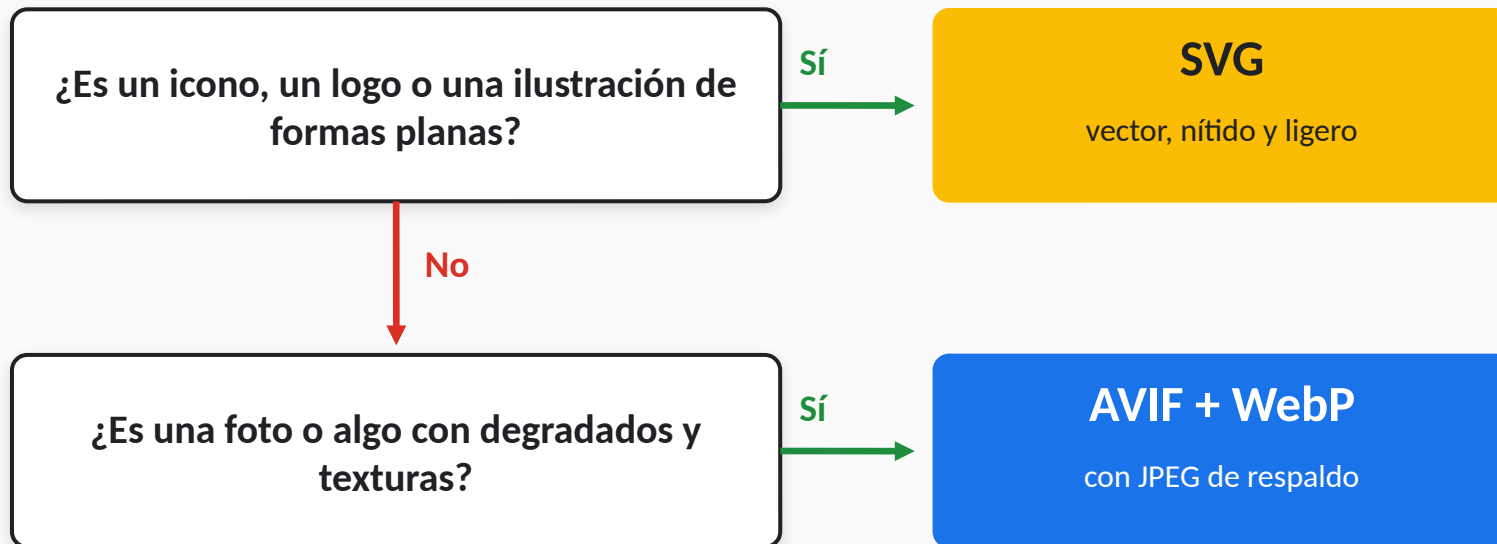
Misma foto, misma calidad visual: JPEG 100 % · WebP ≈ 70 % · AVIF ≈ 50 % del peso (orientativo)



SOPORTE — WebP y AVIF funcionan en todos los navegadores actuales. El JPEG queda como respaldo para los muy antiguos.



¿Qué formato? Dos preguntas



¿Y una captura de pantalla con texto? WebP sin pérdida, o PNG.

En AURIA

Logo: SVG

Iconos (carrito, usuario): SVG en un sprite

Fotos de producto: AVIF + WebP + JPEG

Captura del panel de la app: WebP sin pérdida

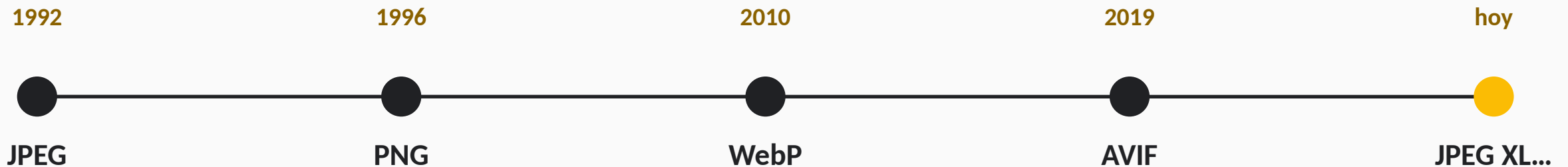
Favicon: SVG (y un PNG de respaldo)



ERROR TÍPICO — PNG para una foto (pesa muchísimo) o JPEG para un logo (bordes sucios y sin transparencia).



El mejor formato de hoy no lo será siempre



JPEG XL

Muy buena compresión y puede convertir JPEG antiguos sin perder nada. Soporte todavía desigual entre navegadores.



Decodificación por hardware

Algunos móviles decodifican ciertos formatos en un chip: gastan menos batería.



Comprobar antes de usar

caniuse.com dice qué navegadores soportan cada formato. El `<picture>` permite probar sin romper nada.



CLAVE — Por eso la estrategia se revisa de vez en cuando: el `<picture>` deja añadir un formato nuevo sin quitar los viejos.



BLOQUE 3

Cada pantalla, su imagen

picture, srcset, sizes, carga diferida y espacio reservado



<picture>: el navegador elige el formato

```
index.html

<picture>
  <source srcset="auria-pro.avif" type="image/avif">
  <source srcset="auria-pro.webp" type="image/webp">
  
</picture>
```

- 1 ¿Entiendes AVIF?
Sí → descarga auria-pro.avif
- 2 Si no: ¿entiendes WebP?
Sí → descarga auria-pro.webp
- 3 Si tampoco
usa el : auria-pro.jpg

Descarga solo UNA: la primera <source> que entiende. No se bajan las tres.

El es obligatorio: es el que se ve, el que lleva el alt y el respaldo para todos.

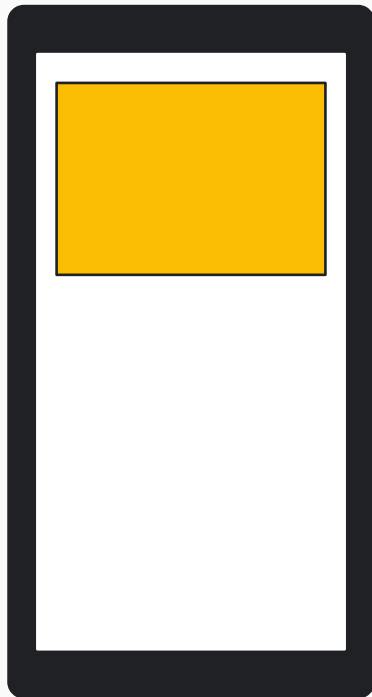
El orden importa: del formato más moderno al más antiguo.



OJO — El CSS se pone al , no al <picture>: el que se pinta en pantalla es el .

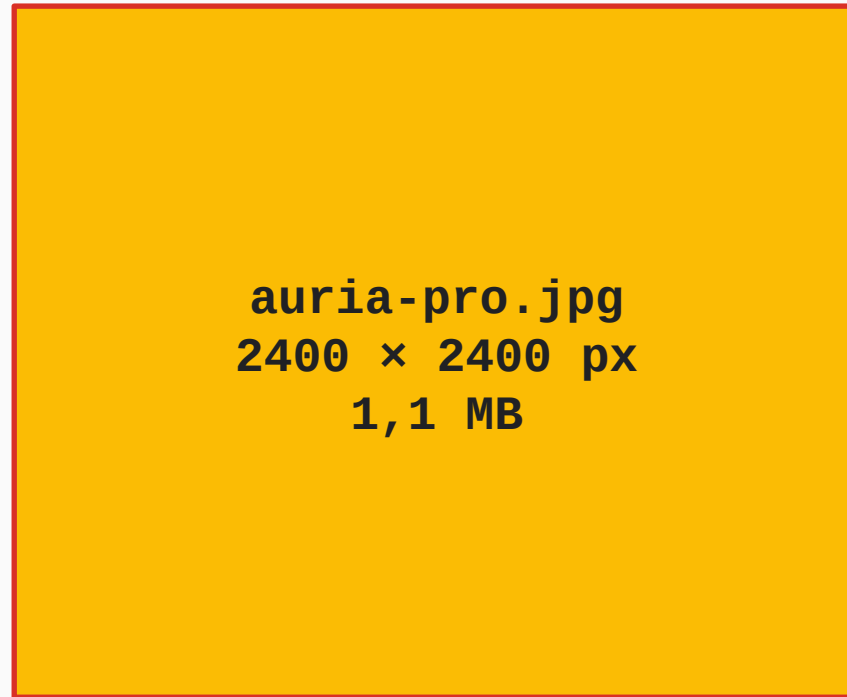


El problema: el móvil se descarga la foto del escritorio



se ve a 390 px

pero descarga



auria-pro.jpg
2400 × 2400 px
1,1 MB

La solución: ofrecer varias

auria-pro-400.webp



28 KB

auria-pro-800.webp



80 KB

auria-pro-1200.webp



160 KB

y que el navegador coja la que necesita



CLAVE — Esto no es el responsive de la UD4: allí cambiaba el CSS. Aquí cambia el FICHERO que se descarga.



srcset y sizes: qué ofreces y cuánto va a ocupar

index.html

```

```

srcset

Los ficheros que tienes y su ancho REAL en píxeles (w = width)

sizes

Lo que va a ocupar en pantalla: desde 48rem, un tercio; si no, todo el ancho

src

El respaldo para navegadores que no entienden srcset

Cómo elige el navegador (un móvil de 390 px con pantalla de densidad 3):

sizes

100vw = 390 px

→

× densidad

390 × 3 = 1170 px

→

elige

la de 1200w

Un portátil normal (densidad 1, ventana de 1366 px) con 33vw necesita ≈ 450 px → coge la de 800w.



OJO — sizes NO cambia el tamaño en pantalla (eso es el CSS): solo le dice al navegador cuánto va a ocupar ANTES de tener el CSS.



Dos casos más: densidad fija y dirección de arte

Tamaño fijo: descriptores x

El logo siempre mide 160 px de ancho. Solo cambia la densidad de la pantalla:

```
header.html


```

Pantalla normal → logo-160 · pantalla retina (2x) → logo-320. Sin sizes. (Si el logo es SVG, ni esto hace falta.)

Dirección de arte: <picture> con media

No es la misma foto más pequeña: es OTRO encuadre para el móvil.

```
hero.html

<picture>
  <source media="(min-width: 48rem)"
          srcset="hero-ancho.webp">
  
</picture>
```

escritorio: panorámica

móvil: recorte



RESUMEN — Mismo encuadre a varios tamaños → srcset + sizes. Encuadre distinto → <picture> + media. Formato distinto → <picture> + type.



loading="lazy": no descargues lo que nadie ha visto

lo que se ve al cargar



lazy: espera

lazy: espera

```
index.html

<!-- La imagen principal: carga YA y con prioridad -->


<!-- Las de más abajo: cuando se acerquen -->

```

loading="lazy": el navegador espera a que la imagen se acerque a la pantalla. También vale para <iframe>.

Nunca en la primera imagen visible: retrasarías justo lo que el usuario está esperando ver.

fetchpriority="high": le dice al navegador que esa imagen es la más importante.



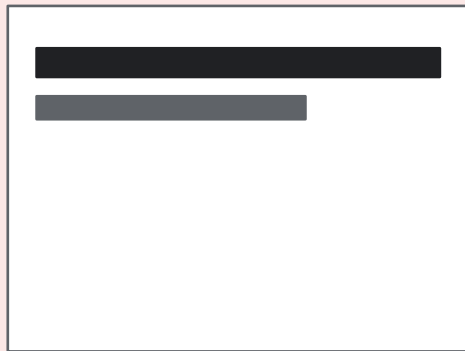
MEDIDA — La imagen grande del principio suele ser el LCP: lo que tarda en aparecer lo más grande de la pantalla (UD12).



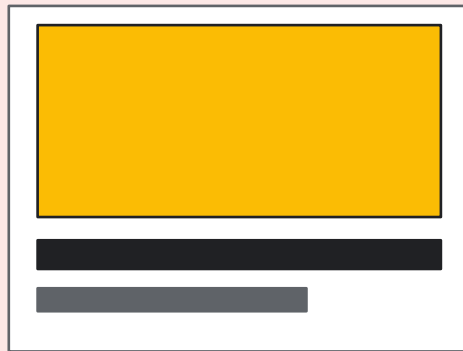
width y height: reservar el hueco antes de que llegue

Sin width y height

cargando...



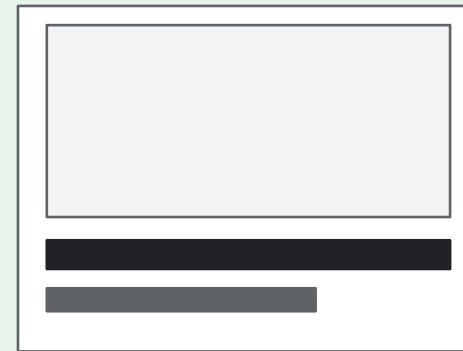
la imagen llega



El texto salta hacia abajo cuando llega la imagen: CLS (UD12)

Con width y height

cargando...



la imagen llega



El hueco está reservado: nada se mueve

```

```

```
img { max-width: 100%; height: auto; }
```



CLAVE — width y height no fijan el tamaño (eso es el CSS): dan la PROPORCIÓN para reservar el hueco. En CSS, height: auto.



El alt: lo que la imagen dice a quien no la ve

Informativa

Aporta contenido

```
alt="Auriculares AURIA Pro en color negro"
```

Describe lo que importa en ese contexto

Decorativa

Solo adorna

```
alt=""
```

alt vacío: el lector de pantalla se la salta.
NUNCA quitar el atributo

Funcional

Es un enlace o un botón

```
alt="Ir al carrito"
```

Describe la ACCIÓN, no el dibujo («icono de carrito»)

Sin «imagen de...» ni «foto de...»: el lector ya anuncia que es una imagen.



UD3 Y UD9 — Ya lo visteis en el HTML semántico. Aquí se aplica a cada imagen nueva; en la UD9 se audita.



BLOQUE 4

SVG y herramientas de imagen

Iconos que se pintan con CSS y el flujo para preparar cada imagen (RA3.c, RA3.d)



Un SVG es HTML que dibuja

```
icono-carrito.svg

<svg xmlns="http://www.w3.org/2000/svg"
      viewBox="0 0 24 24" width="24" height="24"
      fill="currentColor">
  <path d="M7 18a2 2 0 1 0 0 4 2 2 0 0 0 0 0 0-4z..." />
</svg>
```

viewBox

El lienzo: x, y, ancho y alto de las coordenadas internas. El dibujo se escala desde ahí

path d="..."

La forma: órdenes de dibujo (M mover, L línea, C curva, Z cerrar)

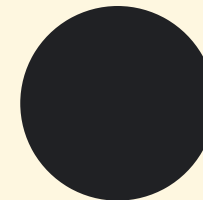
fill="currentColor"

Pinta con el color del texto: el icono cambia de color con CSS (color: ...)

Es texto: se abre con VS Code, pesa bytes y Git ve los cambios.



mismo SVG a 24 px



y a 96 px



OJO — Un SVG exportado de una herramienta trae metadatos y decimales de sobra: se limpia con SVGO (diapositiva 30).



Tres formas de poner un SVG en la página

	<code></code>	SVG en línea <code><svg>...</svg></code>	Sprite: <code><svg><use href></code>
Se cachea	Sí	No (va dentro del HTML)	Sí (si el sprite es un fichero)
Color con CSS	No	Sí (currentColor)	Sí (currentColor)
Repetido 20 veces	1 descarga	20 copias del código	1 definición, 20 <code><use></code>
Animar sus piezas	No	Sí	Limitado
Úsalo para	Logos e ilustraciones	Un icono suelto que anima	Sistemas de iconos



CLAVE — Logo → `<img>`. Iconos de la interfaz → sprite. Un SVG que se anima por partes → en línea.



Sprite SVG: los iconos, definidos una sola vez

img/iconos.svg

```
<svg xmlns="http://www.w3.org/2000/svg">
  <symbol id="carrito" viewBox="0 0 24 24">
    <path d="..." />
  </symbol>
  <symbol id="usuario" viewBox="0 0 24 24">
    <path d="..." />
  </symbol>
</svg>
```



index.html

```
<button class="btn" aria-label="Ver carrito">
  <svg class="icono" aria-hidden="true">
    <use href="img/iconos.svg#carrito" />
  </svg>
</button>
```

<symbol>: define el icono pero no lo pinta. Cada uno con su id.

<use href="...#id">: lo pinta donde lo pongas. Con CSS: `.icono { width: 1.5rem; height: 1.5rem; color: ... }`

aria-hidden="true": el icono es decoración; el nombre accesible lo pone el aria-label del botón.



OJO — Un sprite externo no carga abriendo el HTML con doble clic (file://). Usa Live Server o cualquier servidor local.



BLOQUE 4 • TRATAMIENTO (RA3.C, RA3.D)

Las herramientas: una para cada trabajo



Penpot

Diseño (UD1-UD2)

Exportar iconos y logos a SVG, y piezas a PNG o JPEG



GIMP • Photopea

Edición raster

Recortar, reencuadrar, ajustar color, quitar fondos



Squoosh

Comprimir y comparar

Ver lado a lado JPEG, WebP y AVIF con su peso



SVGO • SVGOMG

Limpiar SVG

Quitar metadatos y decimales de sobra



ImageMagick • sharp

Por lotes

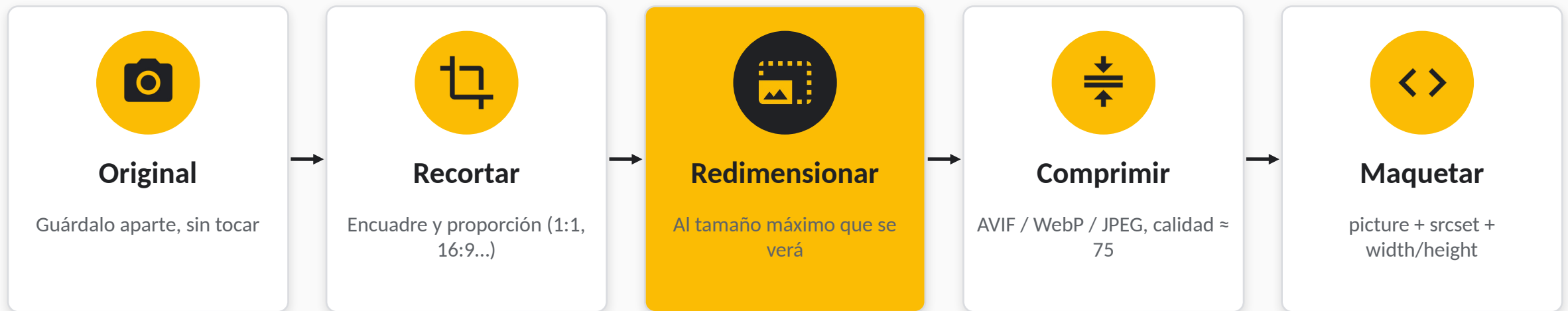
Convertir y redimensionar cien fotos con un comando



TODAS GRATIS — Penpot, GIMP y ImageMagick son software libre; Squoosh, Photopea y SVGOMG funcionan en el navegador.



El flujo: del original a la página



Redimensionar es lo que más ahorra

Una foto de 4000 px mostrada a 400 px tiene 100 veces más píxeles de los que se ven (10×10). Ningún formato compensa eso.



NOMBRES — Nombra los ficheros con su ancho: `auria-pro-800.webp`. En el `srcset` sabrás qué es cada uno sin abrirlo.



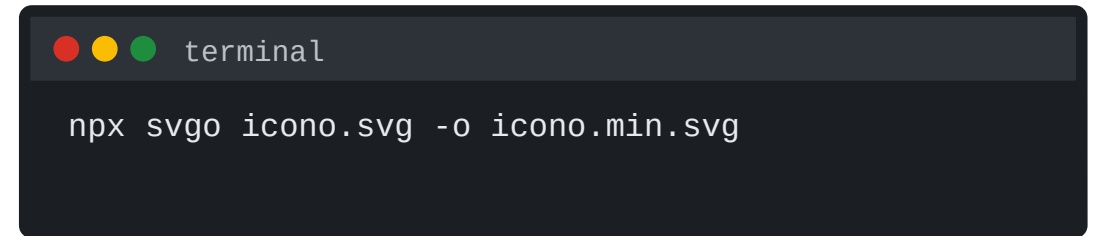
Squoosh y SVGOMG en la práctica

Squoosh (squoosh.app)



- Arrastra la línea: compara las dos mitades
- Cambia formato y calidad: mira el peso abajo
- Haz zoom en bordes y degradados

SVGOMG (y su web SVGOMG)



icono.svg (exportado)



icono.min.svg



Quita comentarios del editor, ids sin usar y decimales de sobra.



CRITERIO — Baja la calidad hasta que notes la diferencia, y sube un poco. No hay un número mágico: depende de cada foto.



Cien fotos no se hacen a mano: por lotes

```
terminal (ImageMagick 7)

# Una foto: redimensionar a 800 px de ancho y guardar en WebP
magick auria-pro.jpg -resize 800x -quality 75 auria-pro-800.webp

# Todas las fotos de la carpeta a 400, 800 y 1200 px
for f in *.jpg; do for w in 400 800 1200; do magick "$f" -resize ${w}x -quality 75 "${f%.jpg}-${w}.webp"; done; done
```



En el build

Vite, Astro o Next generan los tamaños y formatos al compilar



En un CDN de imágenes

Subes un original y el servidor genera la variante que pide cada móvil



En la guía de estilo

Tamaños, calidad y formatos acordados para todo el equipo



CLAVE — La optimización a mano no escala: en un proyecto real se automatiza en el flujo de trabajo.



La guía de estilo también manda en la multimedia

Qué se acuerda	Ejemplo en AURIA
Proporciones	Producto 1:1 • cabecera 16:9 • miniatura de vídeo 16:9
Tamaños y formatos	Fotos: 400 / 800 / 1200 px, AVIF + WebP + JPEG, calidad 75
Tratamiento	Fondo blanco liso en producto, misma luz, sin sombras duras
Iconos	Un solo set (mismo grosor y estilo), sprite en img/iconos.svg, 24 × 24
Vídeo y audio	MP4 H.264 + WebM, siempre con póster y subtítulos
Licencias	Solo CC0, CC BY o bancos con licencia comercial; atribución en /creditos



UD5 — Es la misma guía de estilo de la UD5 con una sección más. Si una foto no la cumple, desentona aunque esté bien comprimida.



BLOQUE 5

Audio y vídeo en la web

Contenedores, códecs, etiquetas HTML5 y herramientas de conversión (RA3.b, RA3.e, RA3.g)



Contenedor y códec: la caja y lo que va dentro

demo.mp4 (contenedor)

vídeo: H.264

audio: AAC

subtítulos, metadatos...

Contenedor: el fichero (.mp4, .webm, .ogg). Mete dentro una o varias pistas.

Códec: cómo está comprimida cada pista (H.264, VP9, AV1, AAC, Opus...).

El navegador necesita entender LAS DOS cosas: un .mp4 con un códec raro no se reproduce.



POR ESO — En `<source>` el type puede llevar el códec: `type='video/webm; codecs="vp9, opus"'`. El navegador sabe si puede antes de descargar.



Los formatos que se usan hoy

Vídeo	Contenedor	Soporte	Notas
H.264	.mp4	Universal	El respaldo: funciona en todo
VP9	.webm	Todos los actuales	Pesa menos que H.264
AV1	.mp4 / .webm	En expansión	El que menos pesa; cuesta más codificar

Audio	Contenedor	Soporte	Notas
MP3	.mp3	Universal	El de siempre
AAC	.m4a / .mp4	Universal	Mejor que MP3 al mismo peso
Opus	.ogg / .webm	Todos los actuales	El mejor para voz y música a poco peso



RECETA — Vídeo: WebM (VP9) + MP4 (H.264) de respaldo. Audio: Opus + MP3 de respaldo. Igual que AVIF + JPEG en imagen.



<video> con varias fuentes, póster y subtítulos

index.html

```
<video controls width="1280" height="720"
  poster="demo-poster.webp" preload="metadata">
  <source src="demo.webm" type="video/webm">
  <source src="demo.mp4" type="video/mp4">
  <track kind="subtitles" src="demo-es.vtt"
    srclang="es" label="Español" default>
  <p>Tu navegador no reproduce vídeo.
    <a href="demo.mp4">Descárgalo</a>.</p>
</video>
```

controls

Los botones del navegador. Sin él, no hay forma de pararlo

poster

La imagen antes de darle a reproducir. Sin ella, a veces se ve negro

<source>

Como en <picture>: el primero que entienda

<track>

Subtítulos en WebVTT: accesibilidad y vídeos sin sonido



OJO — width y height también aquí: reservan el hueco del vídeo, igual que en las imágenes.



autoplay y preload: qué se descarga y cuándo

autoplay: solo en silencio

```
<video autoplay muted loop playsinline ...>
```

- Con sonido, los navegadores lo bloquean
- muted: sin sonido, entonces sí lo dejan
- playsinline: en iPhone no salta a pantalla completa
- loop: vuelve a empezar (fondos, demos cortas)

Y si se mueve más de 5 segundos, botón de pausa (WCAG, UD9).

preload: cuánto adelantar

none

Nada hasta que le den a reproducir

audio, vídeos que casi nadie ve

metadata

Solo duración y tamaño

lo normal en vídeo

auto

El navegador puede bajarlo entero

solo si casi seguro se va a ver



CLAVE — Un vídeo de fondo pesa más que todas las imágenes de la página juntas. Pregúntate si de verdad hace falta.



<audio> y los subtítulos en WebVTT

```
index.html

<audio controls preload="none">
  <source src="podcast.ogg"
    type="audio/ogg; codecs=opus">
  <source src="podcast.mp3" type="audio/mpeg">
</audio>
```

El audio de un podcast o una reseña necesita también una transcripción en texto: quien no oye, o no puede poner sonido.

```
demo-es.vtt

WEBVTT

00:00:01.000 --> 00:00:04.000
Estos son los AURIA Pro.

00:00:04.500 --> 00:00:08.000
Cancelación de ruido activa
y 30 horas de batería.
```

Un fichero de texto: tiempos de entrada y salida, y el texto. Se escribe a mano o se genera y se corrige.



OJO — El type de Opus es audio/ogg; codecs=opus (no «audio/opus»): el contenedor es Ogg.



¿Lo alojas tú o lo incrustas de YouTube?

En tu servidor (<video>)

- Control total del aspecto
- Sin anuncios ni cookies de terceros
- Tú pagas el ancho de banda
- Tú preparas formatos y tamaños

Incrustado (<iframe>)

- La plataforma adapta la calidad a la conexión
- Gratis y sin preparar nada
- Cookies de terceros y anuncios
- El reproductor es el suyo

index.html

```
<iframe src="https://www.youtube-nocookie.com/embed/ID" title="Presentación de AURIA Pro"
width="560" height="315" loading="lazy" allowfullscreen></iframe>
```



DETALLES — youtube-nocookie: sin cookies hasta reproducir. title: lo que anuncia el lector de pantalla. Y loading="lazy".



Herramientas para editar y convertir



Audacity

Audio: cortar, quitar ruido, normalizar volumen, exportar MP3/Opus



HandBrake

Vídeo con ventana: convertir a MP4 o WebM con perfiles para web



FFmpeg

Línea de comandos: lo hace todo y se puede automatizar

terminal (FFmpeg)

```
# A WebM (VP9 + Opus)
ffmpeg -i demo.mov -c:v libvpx-vp9 -crf 32 -b:v 0 -c:a libopus demo.webm
# A MP4 (H.264 + AAC) que empieza a verse antes de descargarse entero
ffmpeg -i demo.mov -c:v libx264 -crf 23 -c:a aac -movflags +faststart demo.mp4
# El póster: un fotograma del segundo 2
ffmpeg -ss 2 -i demo.mov -frames:v 1 demo-poster.webp
```



CRF — -crf es la calidad: cuanto más alto, menos pesa y peor se ve (como el control de calidad de las fotos).



BLOQUE 6

Animar sin vídeo, y medir

Animaciones a partir de imágenes fijas y el peso de la página (RA3.f)



Tres formas de animar a partir de imágenes fijas

@keyframes

Mover, girar, escalar o aparecer una imagen

Ejemplo: Un logo que flota, un icono que late

Sprite sheet

Fotogramas en una sola imagen, que se van mostrando

Ejemplo: Un personaje que anda, un icono dibujado a mano

WebP animado / vídeo

Animaciones complejas hechas en otra herramienta

Ejemplo: Una ilustración animada larga

¿Y el GIF? 256 colores y pesa muchísimo. Un GIF de 3 MB es un MP4 o un WebP animado de 300 KB. Hoy no hay motivo para usarlo.



ORDEN — Primero CSS (no pesa nada), luego sprite, y solo si hace falta un fichero animado.



@keyframes: dar movimiento a una imagen fija

_logo.scss

```
@keyframes flotar {  
  0%, 100% { transform: translateY(0); }  
  50%     { transform: translateY(-10px); }  
}  
  
.logo {  
  animation: flotar 3s ease-in-out infinite;  
}  
  
@media (prefers-reduced-motion: reduce) {  
  .logo { animation: none; }  
}
```

@keyframes

Los momentos de la animación, en %

animation

nombre • duración • ritmo • repeticiones

transform y opacity

Lo único que el navegador anima sin esfuerzo: no anima width, top ni margin

prefers-reduced-motion

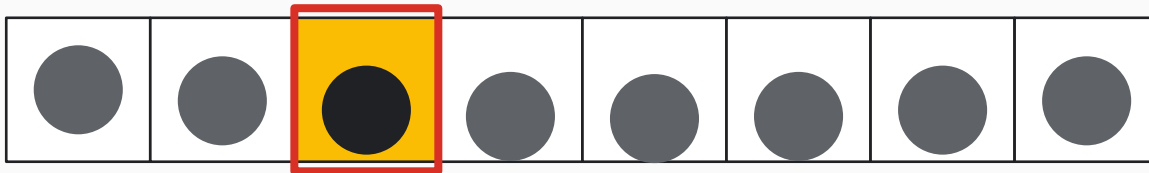
La persona ha pedido menos movimiento en su sistema: se respeta



ACCESIBILIDAD — El movimiento puede marear (trastornos vestibulares). La media query de reduced-motion no es opcional.



Sprite sheet: fotogramas en fila y steps()



pasos.png: 8 fotogramas de $64 \times 64 = 512 \times 64$ px. La «ventana» roja enseña uno cada vez.

_mascota.scss

```
.mascota {  
  width: 64px; height: 64px;  
  background: url("pasos.png") 0 0;  
  animation: andar 0.8s steps(8) infinite;  
}  
@keyframes andar {  
  to { background-position: -512px 0; }  
}
```

Por qué funciona

- El fondo se desplaza de 0 a -512 px (la tira entera)
- steps(8) lo hace a saltos: 8 paradas de 64 px, sin transición entre ellas
- Así cada parada encaja con un fotograma
- Con ease o linear verías la tira deslizarse

Truco: el desplazamiento total = ancho del fotograma × número de fotogramas.



OJO — Con porcentajes en background-position el cálculo cambia y es fácil que los fotogramas no cuadren: usa píxeles.



Medir antes y después: DevTools, pestaña Network

Network	[✓] Disable cache	All	Fetch	CSS	JS	[Img]	Media
hero.avif				avif			142 kB
auria-pro-800.webp				webp			79 kB
auria-mini-400.webp				webp			27 kB
iconos.svg				svg+xml			3 kB

4 / 21 requests 251 kB / 1.2 MB transferred

1

Disable cache

Para medir como la primera visita

2

Filtro Img (y Media)

Solo imágenes, o solo audio y vídeo

3

transferred

Lo que se ha descargado de verdad

4

Throttling: Fast 4G

Simula un móvil con mala cobertura

Presupuesto de peso: un límite acordado, por ejemplo «menos de 500 KB de imágenes en la portada». Si te pasas, se optimiza antes de publicar.



UD12 — Allí se mide con Lighthouse y las Core Web Vitals (LCP, CLS, INP). Hoy basta con saber cuánto pesa y por qué.



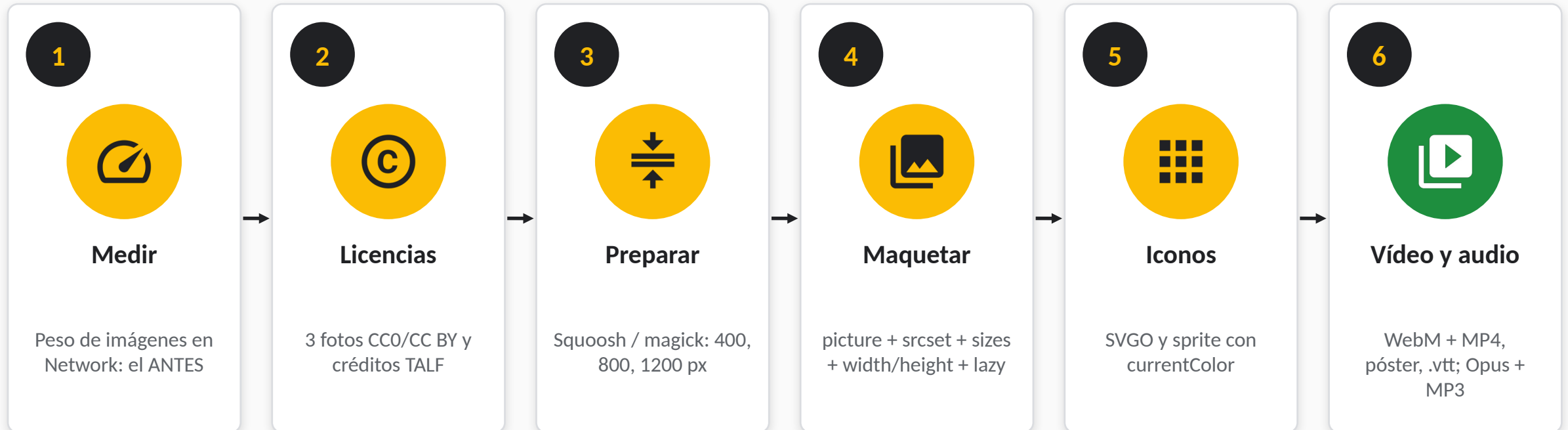
BLOQUE 7

Manos a la obra

Laboratorio AURIA, caso real y errores comunes



La multimedia de AURIA, legal, ligera y medida



Se trabaja sobre la página de AURIA de la UD4-UD5. Al final, volver a medir: el DESPUÉS, con los dos números en el README.



ÉXITO — El peso de imágenes baja de forma medible, no hay saltos al cargar (CLS) y cada recurso tiene su licencia apuntada.



CASO REAL

Multimedia a escala de producción



CDN de imágenes

Cloudinary, imgix o el de Vercel/Netlify: un original y el servidor genera formato y tamaño por dispositivo



Imágenes en el build

Astro, Next o Vite generan srcset, AVIF y WebP solos al compilar



Presupuesto de peso

Límites en KB por página que el equipo se compromete a no pasar



Auditoría continua

La integración continua falla si una imagen pesa más de lo permitido



Impacto en ventas

Las tiendas online documentan más conversión cuando la página carga antes



Créditos y licencias

Los medios guardan la licencia de cada foto: una reclamación cuesta más que la foto



PRUEBA — Abre una tienda grande, F12 → Network → Img: mira qué formato y qué tamaño te sirve en tu pantalla.



Trampas de imagen

PNG para fotografías

Pesa varias veces más sin verse mejor

❌ foto-producto.png (2,4 MB)

✅ picture: avif + webp + jpg

Imagen sobredimensionada

Descargas píxeles que nadie ve

❌ 4000 px para un hueco de 400

✅ srcset 400w, 800w, 1200w + sizes

lazy en la imagen principal

Retrasa lo primero que se ve (LCP)

❌ ``

✅ ``

Sin width y height

La página salta al cargar (CLS)

❌ ``

✅ `<img ... width="800" height="600">`



Trampas de vídeo, audio y licencias

Autoplay con sonido

Molesta y el navegador lo bloquea

✗ `<video autoplay>`

✓ `<video autoplay muted playsinline>`

preload="auto" en todo

Descargas vídeos que nadie verá

✗ `<video preload="auto">`

✓ `<video preload="metadata" poster="...">`

Vídeo sin subtítulos

Inaccesible para quien no oye

✗ `<video><source ...></video>`

✓ `<track kind="subtitles" src="...vtt">`

«La saqué de Google»

Sin licencia no hay permiso

✗ Descargar de un buscador

✓ CC0 / CC BY / banco + créditos



Preguntas de control y debate

1 ¿Por qué SVG es mejor que PNG para el logo de AURIA? ¿Y por qué no sirve para una foto?

2 En un `<picture>` con AVIF, WebP y JPEG, ¿cuántas imágenes descarga el navegador y cuál?

3 ¿Qué indica `sizes` y por qué el navegador lo necesita si el tamaño ya está en el CSS?

4 ¿Por qué la imagen principal no lleva `loading="lazy"`? ¿Qué hacen `width` y `height`?

5 Una foto es CC BY-NC. ¿Puede usarla AURIA? ¿Y una CC BY? ¿Qué hay que poner?



MINI-RETO PRÁCTICO

Reto express: 20 minutos, en parejas

1

Licencia

Una foto CC0 o CC BY y su crédito TALF

2

Tres tamaños

400, 800 y 1200 px en AVIF y WebP

3

picture

Con srcset, sizes, width/height y alt

4

Medir

Peso antes y después en Network

5

Romper

La otra pareja busca saltos y descargas de más

Cómo intentar romperlo

DevTools abierto, Disable cache activado



Pareja A



Pareja B

- Modo dispositivo a 390 px: ¿qué fichero baja?
- Fast 4G: ¿la página salta al cargar?
- ¿El alt dice algo útil? ¿Está el crédito?



ÉXITO — La otra pareja no encuentra saltos de layout ni descargas más grandes de lo necesario.



Licencias, formatos e imágenes en HTML

derecho de autor

Protege la obra desde que se crea, sin registro

Creative Commons

Licencias que dan permisos con condiciones (BY, SA, NC, ND)

CC0

Renuncia a los derechos: uso libre sin citar

raster

Imagen hecha de píxeles: JPEG, PNG, WebP, AVIF

vectorial

Imagen hecha de formas: SVG

con / sin pérdida

Tira detalle para pesar menos / conserva cada píxel

WebP · AVIF

Formatos modernos: mismo aspecto, menos peso

<picture>

Contenedor que ofrece varias fuentes de una imagen

srcset · sizes

Ficheros disponibles · espacio que ocupará

DPR

Densidad de píxeles de la pantalla (1x, 2x, 3x)

lazy loading

Cargar la imagen cuando se acerca a la pantalla

CLS

Saltos del contenido mientras carga la página



Lo demás de la unidad

viewBox

Sistema de coordenadas interno de un SVG

currentColor

El color del texto, usado como relleno del SVG

sprite SVG

Fichero con <symbol>s que se pintan con <use>

SVGO

Optimizador que limpia ficheros SVG

contenedor

El fichero de vídeo o audio: .mp4, .webm, .ogg

códec

Cómo se comprime cada pista: H.264, VP9, AV1, Opus

poster

Imagen que muestra el vídeo antes de reproducirse

preload

Cuánto adelanta el navegador: none, metadata, auto

WebVTT

Formato de texto de los subtítulos (.vtt)

@keyframes

Los pasos de una animación CSS

sprite sheet

Fotogramas en una imagen, mostrados con steps()

LCP

Lo que tarda en verse lo más grande de la pantalla



¿Cuánto le cuesta a quien visita tu web verla?

1

¿Puedo usarlo?

Licencia comprobada y créditos apuntados

2

¿En qué formato?

SVG para formas, AVIF/WebP para fotos, WebM/MP4 para vídeo

3

¿Cuánto pesa?

El tamaño justo para cada pantalla, y medido

Tiempo, datos y batería.

Tú pruebas la web con fibra y un portátil. Quien compra en AURIA quizá lo hace en el autobús, con poca cobertura y pocos datos.

Y el «mejor formato» de hoy cambiará: revisa la estrategia de vez en cuando.



DEBATE — ¿Qué imagen de vuestra AURIA es la que más pesa? ¿Hace falta que pese tanto?

CHECKPOINT

Resumen de puntos clave



Todo tiene dueño: solo se usa con licencia, y se cita como pide



SVG para formas; AVIF y WebP (con JPEG de respaldo) para fotos



picture elige formato; srcset + sizes, el tamaño justo



width/height evitan saltos; lazy, salvo en la imagen principal



Vídeo: WebM + MP4, póster, subtítulos y autoplay solo en silencio



Animar con CSS antes que con ficheros, y respetando reduced-motion

SIGUIENTE • UD7

Componentes UI y Storybook

- Atomic Design: átomos, moléculas, organismos
- La tarjeta de AURIA como componente
- Catálogo aislado con Storybook
- Multimedia dentro de los componentes

Con la UD6 se cierra la 1.ª evaluación. En la UD7 lo hecho hasta ahora se convierte en componentes reutilizables.