

## Cómo servir los artículos HTML para verlos directamente en el navegador

Los HTML autocontenidos (con estilos y scripts embebidos) **sí se ven en el navegador** cuando se sirven por un servidor web con la URL correcta y tipo MIME `text/html`. Servicios como Box o Google Drive suelen ofrecer **descarga** en lugar de vista previa para HTML por seguridad.

### Opciones comparadas

Opción	Ver en navegador	Peso del repo web	Mantenimiento
<b>A. Mismo repo</b> (as-sets/articulos)	En teoría sí; en la práctica el deploy actual devuelve 404 para esos archivos	+3–5 MB por artículo	Un solo repo
<b>B. Repo aparte</b> (ej. cvea-articulos)	Sí, sin depender del build	Repo web ligero	Dos repos
<b>C. Box / Drive</b>	No; suelen forzar descarga	N/A	Solo subir archivos
<b>D. Netlify / Vercel / Cloudflare</b>	Sí	N/A	Subir carpeta o conectar repo

### Recomendación: repo aparte (B)

- **Repo actual (cvea / cvea-platform):** solo la web (Quarto). No subes los HTML aquí.
- **Repo nuevo (ej. cvea-articulos):** solo los `.html` de los artículos. Activas GitHub Pages en ese repo (rama `main`, carpeta `/` o `/docs`). URL tipo: <https://angelucv.github.io/cvea-articulos/articulo-colmenares-espacio-temporal.html>  
Así se ven **directamente en el navegador**, sin descargar.

Ventajas: el repo de la página no se vuelve pesado; los artículos se actualizan sin tocar el build de Quarto; enlaces estables y fáciles de cambiar en la web.

## Tamaño aproximado

- Artículo Colmenares (espacio-temporal): ~3,4 MB
- Artículo Moreno-Colmenares (mortalidad): ~4,6 MB

Un repo con 10–20 artículos similares podría rondar 50–100 MB; GitHub lo admite. Si todos estuvieran en el repo de la web, el clone y el build serían más pesados; por eso es mejor el repo separado.

## Pasos para el repo «cvea-articulos»

1. Crear en GitHub un repo nuevo (ej. `cvea-articulos`), vacío o con un `README`.
2. En tu máquina está la carpeta `cvea-articulos-repo` (en la raíz del proyecto) con los dos HTML y un `README`; esa carpeta no se sube al repo de la web (está en `.gitignore`). Úsala para el nuevo repo:
  - `cd cvea-articulos-repo`
  - `git init`
  - `git add .`
  - `git commit -m "Artículos RVEA HTML"`
  - `git remote add origin https://github.com/TU_USUARIO/cvea-articulos.git`
  - `git branch -M main`
  - `git push -u origin main`
3. En el repo en GitHub: **Settings** → **Pages** → **Source**: rama `main`, carpeta **Root**.
4. La web CVEA ya tiene los enlaces apuntando a `https://angelucv.github.io/cvea-articulos/...`; en cuanto el repo exista y Pages esté activo, se verán los artículos en el navegador.

Cuando tengas la URL base del repo (ej. `https://angelucv.github.io/cvea-articulos/`), los enlaces serían:

- Colmenares: `https://angelucv.github.io/cvea-articulos/articulo-colmenares-espacio-temporal.html`
- Moreno-Colmenares: `https://angelucv.github.io/cvea-articulos/articulo-moreno-colmenares-mortalidad-infantil.html`