

Hacer que mi blog sea privado

Miguel

Supongamos que quieres hacer que sólo los coautores del trabajo tengan acceso al *blog* o a alguna sección del mismo. Esto puede iniciar procurando que los *bots* o *arañas* de *Google* y otros buscadores no recojan datos o metan las narices en mi contenido. Bueno, esto puede hacerse por dos caminos. Lo primero lo haremos con el concepto **noindex** y la ayuda de instrucciones directas a los *bots* mediante un archivo de indicaciones. El resultado es poco invasivo para el usuario pero no enteramente seguro para garantizar la privacidad del contenido. Una solución más radical es de plano encriptar el sitio o porciones sensibles del mismo, para sólo mostrarlo cuando se proporcionen una clave específica. Un archivo encriptado no es indexable (es parte de la *Deep web*, que no es lo mismo que la *Dark Web*). Este último método es muy efectivo, pero claro, potencialmente un poco incómodo para el usuario. Desde luego, las fugas de información son posibles, pero más bien vinculadas al cuidado que tengan los involucrados en la gestión del acceso y del manejo del contenido. Si todo esto es necesario en un marco de *ciencia abierta* es algo que vale la pena meditar, pues el acceso temprano es congruente con las nociones de *pre-registro* y *pre-publicación*.

No index

Lo que hay que hacer para evitar a los *bots* en la página principal, es poner en el archivo `_quarto.yml` las siguientes líneas:

💡 Archivo `_quarto.yml`

```
project:
  type: website
  resources:
    - robots.txt

format:
  html:
    theme: fotoD.png
    include-in-header:
      - text: <meta name="robots" content="noindex">
```

①

②

- ① Hay que proporcionar una lista de instrucciones para los robots que andan cosechando por la Web. Se hace mediante un archivo específico como el anotado.
- ② Esta es tu petición de no indexar por favor. Sigue el código concreto que lo indica, que debe aparecer en la sección **head** del sitio Web que contiene tu *Blog*.

Con esto evitaras que muchos buscadores, los más formales, te den gusto y no reúnan información de tu sitio. No todos los buscadores cumplen con este estándar, pero sí lo hacen los más importantes. Esto también te hace ver que es un *acto voluntario*, es decir, un metiche puede optar por ignorar tus deseos.

Las instrucciones para los robots pueden variar. En este caso puse las siguientes en el archivo `robots.txt` en la raíz de tu proyecto.

💡 Archivo robots.txt

```
robots.txt generated by www.seoptimer.com
User-agent: Googlebot
Disallow: /
User-agent: googlebot-image
Disallow: /
User-agent: googlebot-mobile
Disallow: /
User-agent: MSNBot
Disallow: /
User-agent: Slurp
Disallow: /
User-agent: Teoma
Disallow: /
User-agent: Gigabot
Disallow: /
User-agent: Robozilla
Disallow: /
User-agent: Nutch
Disallow: /
User-agent: ia_archiver
Disallow: /
User-agent: baiduspider
Disallow: /
User-agent: naverbot
Disallow: /
User-agent: yeti
Disallow: /
User-agent: yahoo-mmccrawler
Disallow: /
User-agent: psbot
Disallow: /
User-agent: yahoo-blogs/v3.9
Disallow: /
User-agent: *
Disallow: /
Disallow: /cgi-bin/
```

Una solución más definitiva es encriptar tu contenido. En este caso te sugiero usar la solución que se ofrece con la biblioteca **staticcryptR**. Las [explicaciones de cómo y qué instalar están acá](#).

Si la tradicional instalación de librerías no es suficiente, te sugiero ir al documento sugerido y ver las líneas que hablan de *Node.js*, un ambiente de *javascript* que agrega una capa de procesamiento necesario para esto en tu máquina. Luego, asegúrate de tener instalada en **R** la biblioteca mencionada. Con eso hecho. Ve de nuevo a tu archivo `_quarto.yml` y agrega las líneas necesarias para que incluya lo que se indica:

💡 Orden de encriptar en `_quarto.yml`

```
project:
  type: website
  output-dir: "_site"
  post-render: encrypt.r
  resources:
    - robots.txt
```

①
②

- ① Indica el archivo o directorio que quieres encriptar, en este caso es **todo el sitio**, qué vive en el directorio `_site`.
- ② Anota el archivo que contiene las instrucciones concretas sobre qué y cómo encriptar el contenido.

Desde luego, también necesitas el código indicado que pondrás en el archivo `encrypt.r` y que alojaras en la raíz de tu proyecto.

💡 Código para crear el archivo `encrypt.r`

archivo `encrypt.r`

```
staticcryptR::staticcryptr(
  files = "_site/",
  directory = ".",
  password = "pita-123",
  short = TRUE,
  recursive = TRUE
)
```

①
②
③
④
⑤

- ① Directorio objetivo, en este caso: todo el sitio.
- ② En dónde encontrar el directorio objetivo. En este caso la raíz de tu proyecto.
- ③ La clave que usaras para la tarea (sólo una), puedes usar **keyring** para ocultarla.
- ④ Indicas que es válido usar claves cortas (quizás sólo 8 caracteres). Es preferible claves largas.
- ⑤ Especifica que quieres encriptar la carpeta indicada y todas las subcarpetas que contenga.

Cuidado

Al encriptar `_site` los paquetes ya codificados se van “acumulando”, lo que se traduce en que tengan un tamaño cada vez mayor, hasta volverse inmanejables. Para evitar ese efecto no deseado, es conveniente usar preferentemente la opción **Render Website** desde la pestaña **Build** en *RStudio*. Esto recrea todos los archivos desde cero. Archivos muy grandes acaban siendo un obstáculo para el manejo eficiente en *git*. Para ayudar a visualizar el problema agregué unas líneas en el escript de `encrypt.r` que te ayudarán a visualizar si están apareciendo archivos potencialmente problemáticos, es decir mayores a 10MB. La advertencia la podrás ver en la pestaña *Background Jobs*, pero sólo aparecerá si se detecta algún riesgo.

💡 Vigilar tamaño de archivos en `_site`

Sí te interesa mantener la vigilancia del tamaño de los archivos en `_site`, este es un código que lo hace y te reporta cuando se sospecha de algún riesgo por archivos mayores a 10MB.

```
lista_rchivos <- tibble::tibble(archivo = list.files("./_site/",
                                                    recursive = TRUE,
                                                    full.names = TRUE)) |>
  dplyr::mutate(MB = file.size(archivo)/(1024^2)) |>
  dplyr::filter(MB > 10)

if (dim(lista_rchivos)[1] > 0)
{
  print("Advertencia: hay archivos muy grandes en _site")
  print("Te sugiero hacer un render completo desde la pestaña 'Build'")
  print(lista_rchivos)
}
```