

Manual para publicar un libro en GitHub creado en Bookdown

Alvaro Martínez Rodríguez

2022-01-07

Antes de empezar	2
Cuenta y repositorio	2
Programa Git	3
Preparar la carpeta del proyecto	5
Carpeta docs	5
Archivos de texto	5
Subida de archivos	8
Vinculaciòn	8
Link	11
Actualizaciòn	13

Cuenta y repositorio

Para poder hacer una publicación en **GitHub** es necesario crear una cuenta en esta plataforma. Para ello, hay que entrar en github.com y dar clic en *Registrarse* y posteriormente entrar a la cuenta. Una vez dentro, en el lado superior izquierdo se encuentra el boton *new* para crear un nuevo repositorio¹.

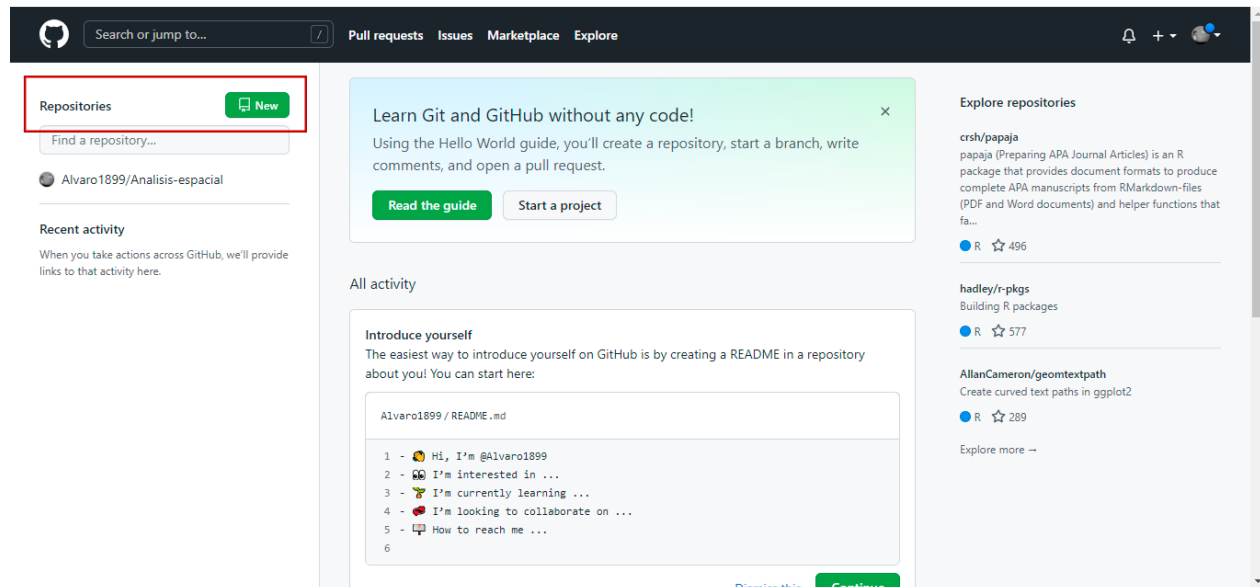


Figura 1: .

Para crear el repositorio solo es necesario darle un nombre y dar clic en el boton *Crear repositorio*. Por ahora será todo en GitHub, pero dejaremos la pagina abierta.

¹Un repositorio es el sitio dentro del cual se alojarán los archivos del libro

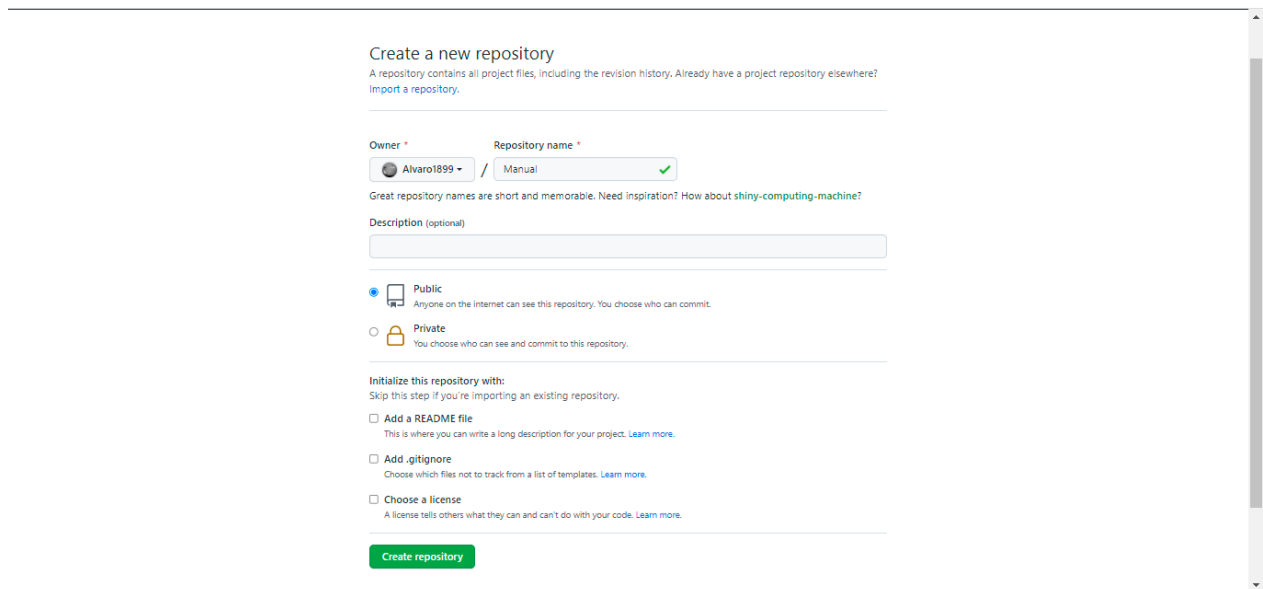


Figura 2: .

Programa Git

Por otro lado tambien es necesario haber descargado Git. Este programa nos ayudará a enlazar R studio con GitHub. Damos clic en descargas para windows y después ejecutamos el archivo Git-2.34.1-64-bit.exe que se descargó para instalar Git.

Por último, es necesario entrar en nuestro proyecto de R studio y seguimos los siguientes pasos.

Abrimos *Tools* en la cinta de opciones -> *Version Control* -> *Project Setup* -> Opción *Git/SVN* -> Cambiamos la opción de *control project system* de **None** a **Git**

Después de realizar los pasos anteriores iremos a *View* en la cinta de opciones y damos clic en *Show Git*. Con ello nos aparecerá la pestaña Git que usaremos más adelante.



Figura 3: .

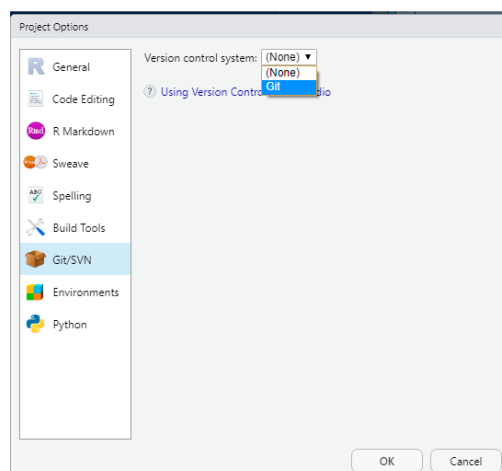


Figura 4: .

PREPARAR LA CARPETA DEL PROYECTO

Carpeta docs

Ahora, el siguiente paso es preparar la carpeta de nuestro proyecto. En este punto se espera que el lector ya haya construido el libro a través del botón *Build Book* en formato **bookdown::github** o **bookdown::bs4_book** de manera correcta.

En nuestra carpeta de proyecto debemos crear una carpeta con el nombre de *docs*, en ella, GitHub podrá leer los archivos necesarios para el libro. Una vez creada, entraremos en la carpeta **_book**. Aquí se encuentran los archivos renderizados en html de cada capítulo, así como todos los recursos que utilizan. Todos estos archivos los copiaremos y los pegaremos en la carpeta que creamos.

Archivos de texto

Lo siguiente a realizar es la creación de 4 archivos: *.nojekyll*, *.Rbuildignore*, *.travis.yml*, *DESCRIPTION*

El primer archivo lo crearemos dentro de la carpeta *docs* y para ello abriremos un nuevo archivo en R de tipo texto. No escribiremos nada en el archivo, solo lo guardamos con el nombre **.nojekyll**

File -> New File -> Text File

Volvemos a crear un archivo de texto, escribimos el siguiente código dentro y lo guardamos en la carpeta principal con el nombre **.Rbuildignore**

Ahora creamos otro archivo de texto con el siguiente código y lo guardamos con el nombre *.travis.yml*

Por último creamos otro archivo con el nombre de **DESCRIPTION** y escribimos lo siguiente:

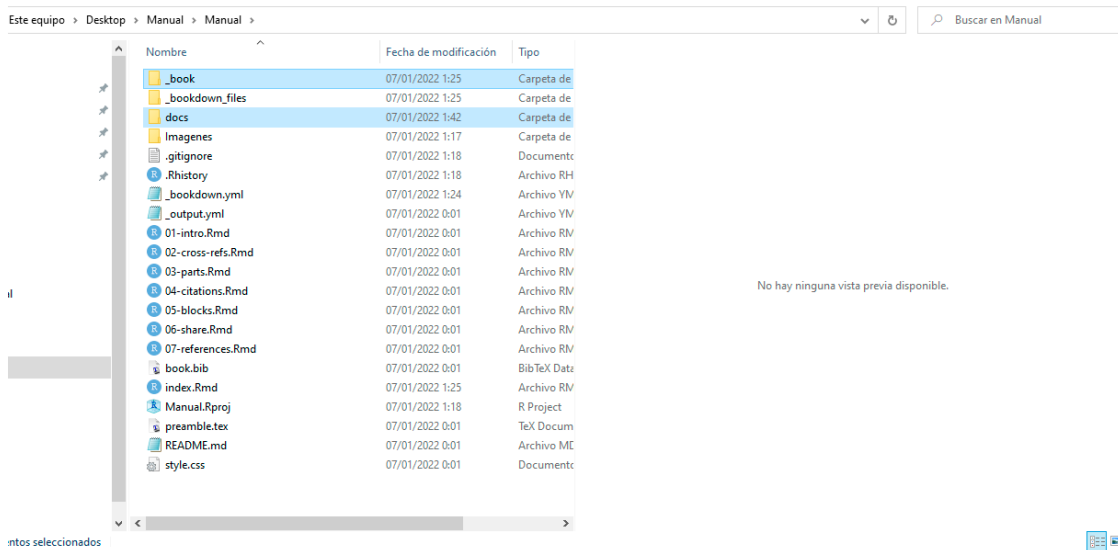


Figura 5: .

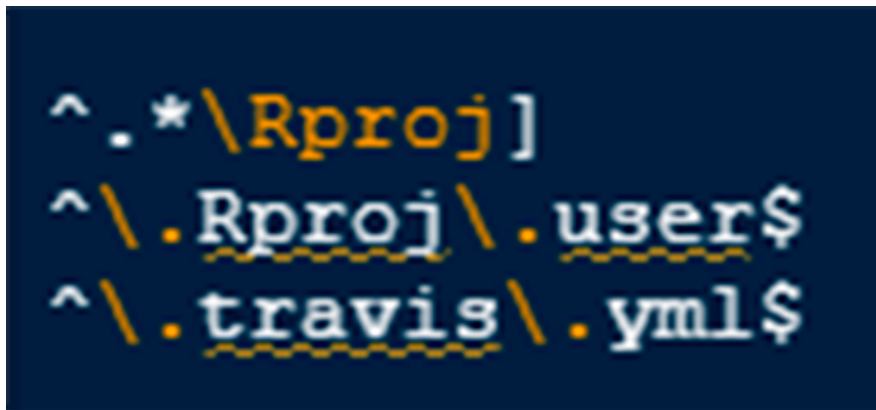


Figura 6: .

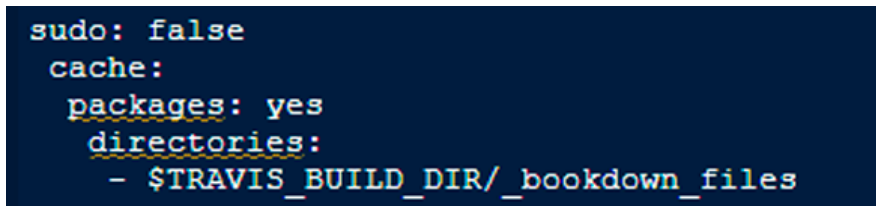


Figura 7: .

```
Package: placeholder  
Type: Book  
Title: Analisis espacial con R  
Version: 0.0.1  
Remotes: rstudio/bookdown
```

Figura 8: .

Vinculación

Hasta ahora hemos preparado nuestro proyecto para poderlo subir de manera correcta. Iremos a R studio para comenzar a subir los archivos. En la pestaña de *Git* y en el botón de engrane seleccionaremos la opción *Shell*. Se abrirá una ventana cmd, en ella, deberemos copiar el código que aparece en nuestro repositorio creado de GitHub. Esto permitirá que Rstudio y GitHub este conectados y le indicara a R en donde debe subir los archivos.

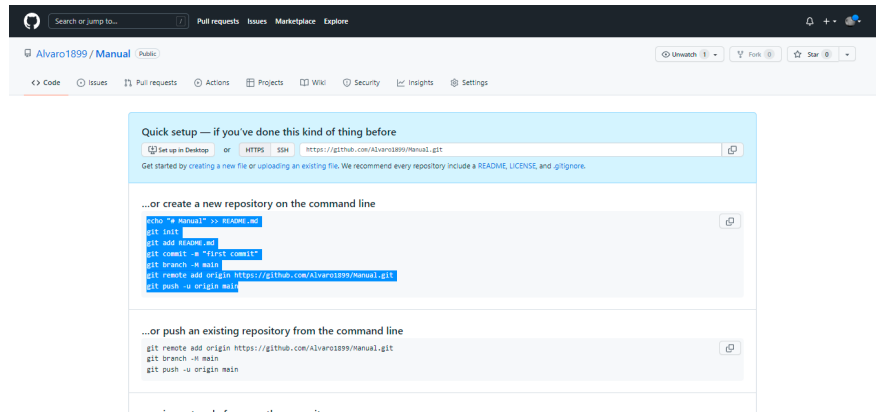


Figura 9: .

Cerramos la ventana de sistema cmd y regresamos a Rstudio. En la misma pestaña de Git nos aparecerán una serie de archivos que contiene nuestro libro y deberemos seleccionarlos todos dando clic en la columna de *Staged*².

Presionaremos el botón de *Commit* y escribiremos en el recuadro derecho “Subida de archivos” y daremos clic en el botón *Commit*. Una vez termine el proceso resultante de paso anterior daremos clic en la flecha verde *Push* que esta en la parte superior derecha y esperaremos a que se suban los archivos.

²En ocasiones, después de dar clic en el recuadro blanco tarda unos segundos en marcarlo, por lo que deberemos ser pacientes y no dar más de un clic por archivo.

```
C:\Windows\system32\cmd.exe

C:\Users\Jarvis\Desktop\Manual\Manual>git init
Reinitialized existing Git repository in C:/Users/Jarvis/Desktop/Manual/Manual/.git/

C:\Users\Jarvis\Desktop\Manual\Manual>git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

C:\Users\Jarvis\Desktop\Manual\Manual>git commit -m "first commit"
[main ec714db] first commit
1 file changed, 1 insertion(+)

C:\Users\Jarvis\Desktop\Manual\Manual>git branch -M main

C:\Users\Jarvis\Desktop\Manual\Manual>git remote add origin https://github.com/Alvaro1899/Manual.git
error: remote origin already exists.

C:\Users\Jarvis\Desktop\Manual\Manual>git push -u origin main
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 4 threads
Compressing objects: 100% (26/26), done.
Writing objects: 100% (28/28), 651.98 KiB | 9.45 MiB/s, done.
Total 28 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/Alvaro1899/Manual.git
 * [new branch]    main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

C:\Users\Jarvis\Desktop\Manual\Manual>
```

Figura 10: Resultado de pegar el código

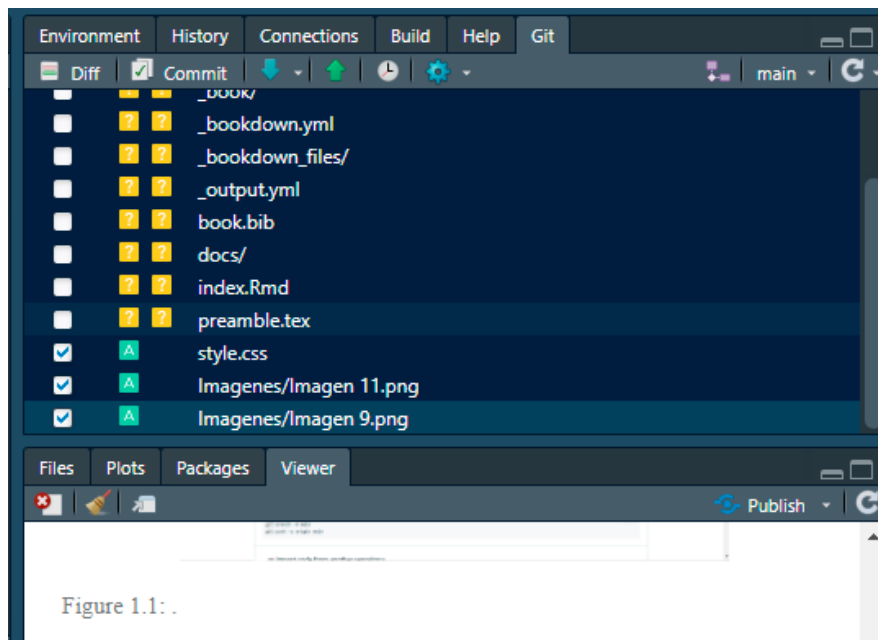


Figura 11: .

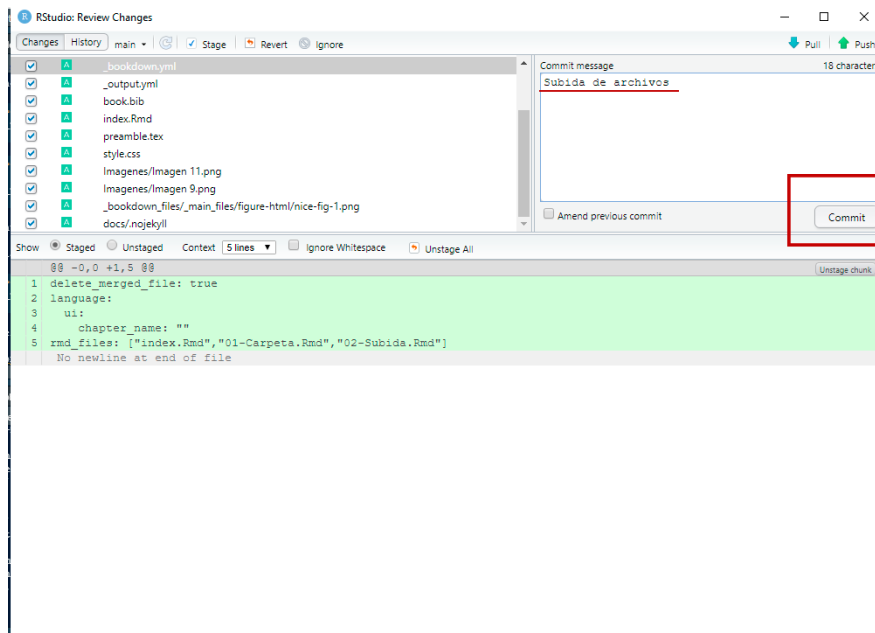


Figura 12: .

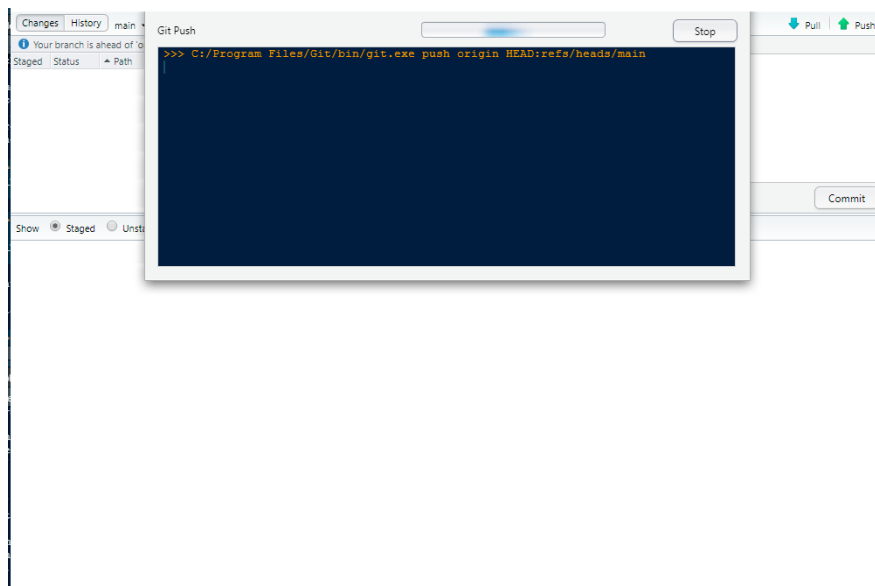


Figura 13: Resultado final de *Push*

Ahora ya tenemos nuestros archivos subidos en nuestro repositorio de GitHub.

Link

Para obtener el link del documento que acabamos de publicar debemos ir a nuestro repositorio en GitHub. Una vez dentro, podremos observar en la pantalla principal todos los archivos que se han subido de nuestro libro. Daremos clic en *Settings*

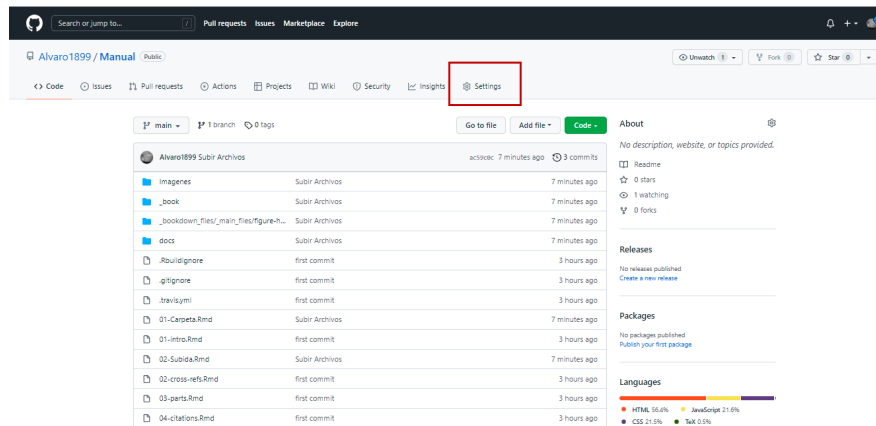


Figura 14: .

Dentro de *Settings* nos aparecerá un menú a la izquierda, en donde seleccionaremos la opción de *Pages*. En este apartado tenemos la sección de *Source*, ahí debemos cambiar la opción **None** por la rama **Main** y la carpeta **Docs** y lo guardamos con el botón *Save*. Una vez lo anterior, nos aparecerá en la parte de arriba el link de nuestro documento publicado.

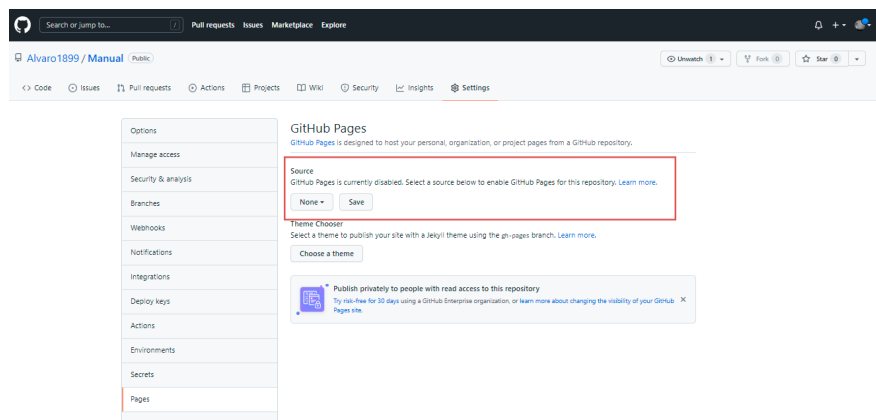


Figura 15: .

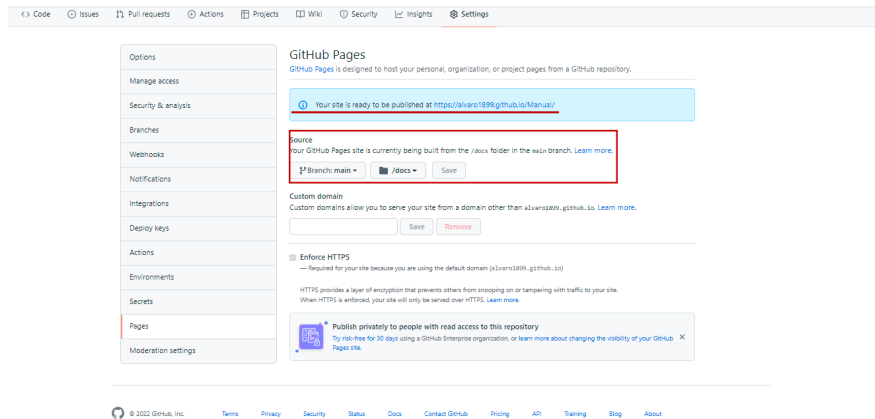


Figura 16: .

Para actualizar nuestro libro solo hace falta volverlo a correr en formato **bookdown::github** o **bookdown::bs4_book**. Esto generara automáticamente los archivos en la carpeta `*_book`. *Copiamos los archivos generados en docs, vamos a la carpeta Git en R, seleccionamos todos los archivos que aparezcan, entramos en Commit, escribimos “Actualización”, presionamos en Commit* de nuevo y después en Push.* Verificamos que se hayan actualizado los documentos en GitHub.