

Enlazar Github con Netlify para publicar

Miguel Equihua

Xalapa, Ver., 3 de julio, 2024

Preparación

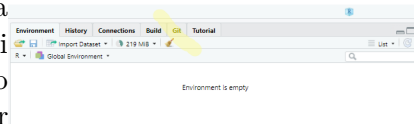
Supondré que ya has obtenido tus cuentas de *Github* y *Netlify*. En todo caso y por si las dudas estos dos enlaces te deben ayudar.

💡 Enlaces para registro en Github y Netlify

[Github](#)

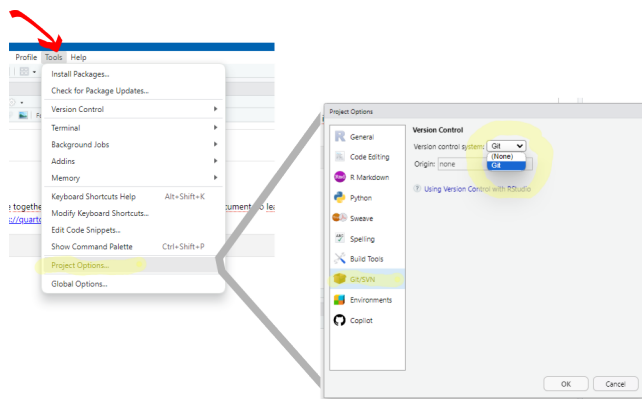
[Netlify](#)

Ya tenemos un proyecto *Quarto* en *RStudio*, al que podemos agregar la funcionalidad de control de versiones mediante *Git*. Por otro lado, también podemos iniciar un nuevo proyecto que desde un inicio tenga esta característica, la [forma e hacerlo está aquí](#). Si prefieres el camino de agregar *Git* a tu proyecto ya existente habrá que habilitarlo. Para saber si está *Git* habilitado basta con averiguar si tienes la pestaña respectiva en el cuadrante de *ambiente de trabajo* de *RStudio* (arriba a la izquierda normalmente)



💡 Agregar control de versiones a tu proyecto

Camino A: usando el menu



Es muy probable que por este camino tengas una rama principal llamada *master*. Aunque no hay ningún inconveniente con esto, actualmente se prefiere usar como rama principal una llamada *main*. Si esto es lo que prefieres tu también puedes usar el siguiente comando desde la pestaña de consola.

```
usethis::git_default_branch_rename(from = "master", to = "main")
```

Camino B: usando comandos

1. Asegúrate de tener instalada la biblioteca `usethis`
2. Ejecuta el siguiente comando en *R* y responde las preguntas que aparecerán en la consola

```
usethis::use_git()
```

Agregar el nuevo repositorio a Github

Lo más sencillo es de nuevo usar la biblioteca de ayuda `usethis`. En la consola ejecuta el siguiente comando

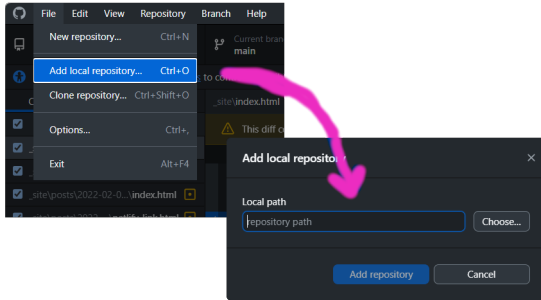
```
usethis::use_github()
```

Lo que ocurre con este comando es lo siguiente:

1. Crearás un nuevo *repositorio en GitHub* con el mismo nombre de tu proyecto.

2. Configurarás el nuevo *repo* como el *origin* remoto para tu *repo* local.
3. Prepararás una rama local inicial a la que nombrarás **main** y sobre la que se hará el seguimiento hasta nueva orden.
4. Sincronizarás lo que tengas en tu máquina con *Github*.
5. Abrirás el nuevo repositorio en el navegador para que puedas ver lo que has hecho.

Casi tan simple como lo que acabamos de hacer es utilizar *Google Desktop* para crear un *repo* en tu cuenta, con base en un proyecto de tu máquina. Asegúrate de hacer las tareas 3 y 4 anotadas arriba.



Un último paso sólo para asegurar un poco más tu espacio de trabajo y colaboración es recurrir al comando siguiente.

```
usethis::git_vaccinate()
```

Con esta instrucción se revisa el archivo `.gitignore` global que especifica lo que no conviene, se quiere o debe compartir (claves confidenciales por ejemplo), en un espacio abierto.

De ahora en adelante puedes tomar registro de tu avance de trabajo, involucrando todos los directorios de tu proyecto (salvo los que estén en `.gitignore`), cada vez que lo consideres conveniente (al dar por terminada una tarea propuesta, por ejemplo). También podrás sincronizar este registro con el repositorio que has creado en *Github* cuando te parezca prudente. Toma nota de que nada de esto ocurrirá automáticamente, deberás generar la indicación cada vez que quieras que ocurra.

La rutina de operación con *Git* se ilustra en la siguiente figura.

En caso de Incendio

 **1. git commit**

 **2. git pull**

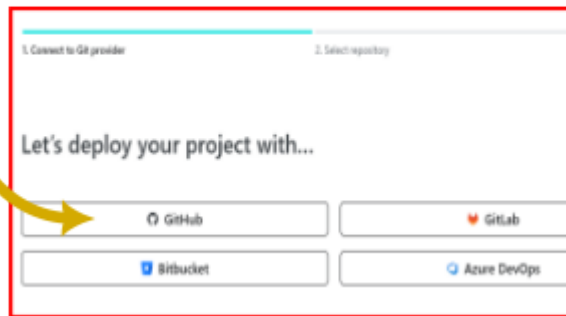
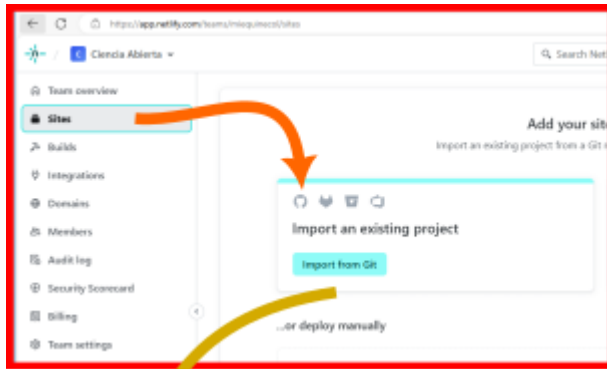
 **3. git push**

 **4. git out**

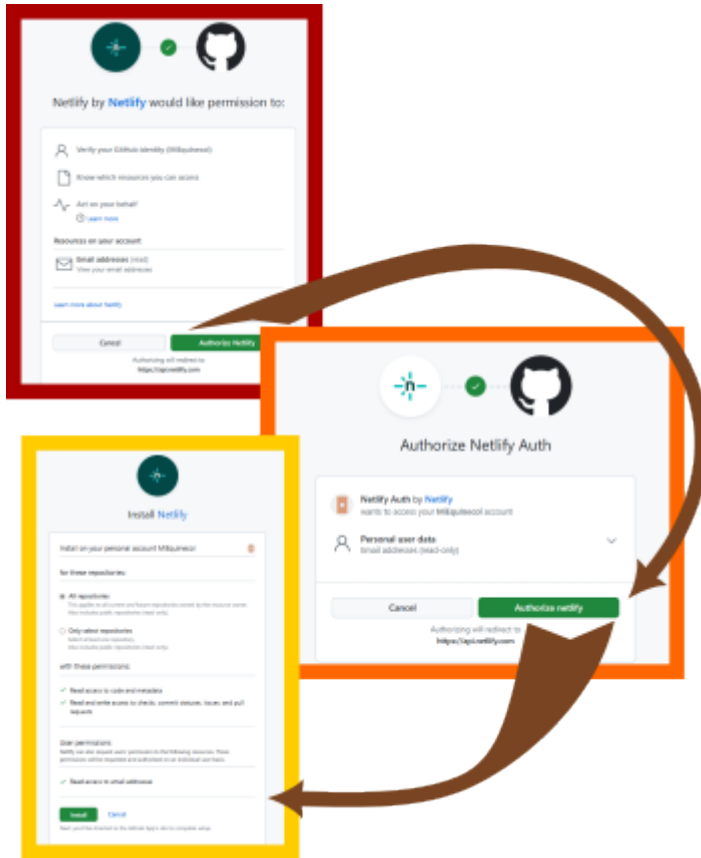
Preparar Github para interactuar con Netlify

Hay que preparar a *Github* dando acceso a *Netlify* para que tome lo necesario. La meta es que construya un sitio Web con tu contenido y lo publique en Internet. Los pasos que hay que seguir para esta primera interacción son los siguiente.

Iniciar la vinculación con Github seleccionando la opción qe ofrece importar los documentos desde n *repo* Git. Esto dará la opción de utilizar *Github* como origen de datos, entre otras posibilidades.



El siguiente paso es autorizar a Netlify a acceder a Github a través de tu cuenta, así como los específicos del repositorio que te interesa vincular. Esto también implica instalar una **aplicación** de vínculo entre *Netlify Github* dentro de tu cuenta.

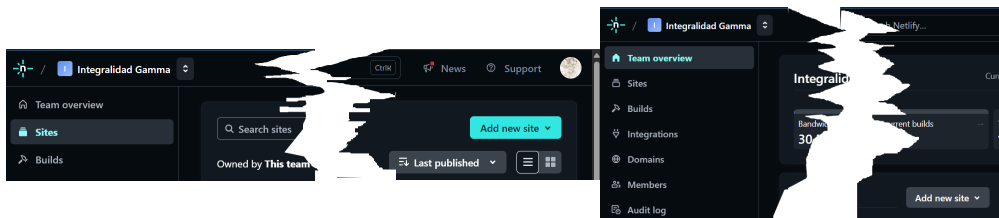


Si todo ocurrió sin problemas, tendrás ahora en *GitHub*, en el menú de aplicaciones (Avatar → Settings → Applications), un botón que te permitirá configurar el vínculo con Netlify según tus requerimientos. También podrás ver los *repos* que hayas autorizado desde *Netlify*.

Creación de un nuevo sitio a publicar

1. En Netlify:

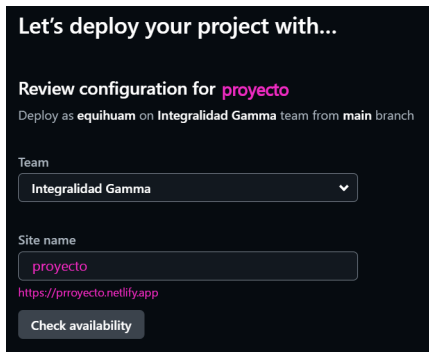
- Desde la opción *team* o *site* puedes generar un nuevo sitio.



Cuando aprietas el boton de *añadir sitio*, aparecerá una nueva pantalla que tiene tres secciones. Se trata de los atributos que tendrás que proporcionar para darle presencia en Internet a tu proyecto y algunos otros atributos que definen como se producirá y actualizará continuamente. Estas operaciones es poco probable que las vuelvas a ver, una vez que tu proyecto esté en producción, aunque desde luego estarán siempre ahí por si deseas hacer ajustes.

¿Qué nombre le daras?

Deberás elegir un nombre que se convertirá en una URL para acceder a tu proyecto. Puede ser cualquier cosa que desees, pero debe ser único. En esta sección puedes escribir nombres y verificar que estén disponibles



¿Qué hará Netlify para operar tu sitio?

Es una colección de atributos para indicarle a *Netlify* dónde conseguir los documentos y como manejarlos. En nuestro caso, muy simple, básicamente hay que decirle en donde están los documentos que *Quarto*, con ayuda de *pandoc*, a *renderizado*. Si no has cambiado nada en `_quarto.yml` la rama que estamos usando aquí para que *Git* los registre es **main** y, en ella el directorio de producción se llama `**_site**`. Por favor verifica el contenido de esto para ayudarte a comprender mejor lo que estás haciendo.

Build settings
Specify how Netlify will build your site.
[Learn more in the docs](#) ↗

Branch to deploy
main

Base directory
The directory where Netlify installs dependencies and runs your build command.

Build command
Examples: jekyll build, gulp build, make all

Publish directory
Examples: _site, dist, public

Functions directory
Example: my_functions

¿Todo listo? ¡a producción!

En nuestro caso no hay más que hacer, *Netlify* tiene información suficiente para encargarse de publicar tu proyecto continuamente. Incorporará los cambios que hagas en *RStudio* en la rama principal. Lo hará automáticamente cada vez que envíes tus cambios a *GitHub*.

Environment variables
Define environment variables for more control and flexibility over your build.

Add environment variables

Deploy **netlify**

Si todo salió bien, en este momento ya debe estar tu proyecto publicado y accesible para cualquier lector del mundo que lo localice y se interese en su contenido.