

Guía sobre Github

Repositorios online

Para poder colaborar en cualquier proyecto [Git](#), necesitamos saber cómo gestionar nuestros repositorios remotos. Los repositorios remotos son versiones de tu proyecto que se encuentran alojados en Internet o en algún punto de la red. Colaborar con otros implica gestionar estos repositorios remotos, y mandar (push) y recibir (pull) datos de ellos cuando necesitemos compartir cosas.

Hay muchos servicios que permiten guardar repositorios de manera online como GitHub, Bitbucket o GitLab entre otros.

Nosotros vamos a usar el más usado y famoso, GitHub!

GitHub

Lo primero que tenemos que hacer es [registrarnos](#).

GitHub es una plataforma de desarrollo colaborativo, es gratuita y te permite guardar repositorios de git para trabajar de manera distribuida solo o con tu equipo.

Nuevo repositorio

Para ver como funciona vamos a crear un nuevo repositorio personal.

1- Arriba a la derecha, haciendo click en el signo + y opción New Repository vamos a ir al menú de nuevos repositorios.



2- Seleccionamos el nombre para nuestro repositorio.

3- Los repositorios privados solo van a ser visibles por el dueño y sus colaboradores.

Podemos agregar una licencia para nuestro proyecto o un `.gitignore` que te permite agregar un listado de archivos que quieras que git ignore a la hora de guardar tus cambios, pero no es importante que se preocupen por eso por ahora.

4- Create Repository para finalizar.



Vinculando repositorios

Una vez creado el repositorio, al estar vacío, git nos da unas recomendaciones de como arrancar, nosotros vamos a optar por la opción de crear un repositorio local y vincularlo al remoto.

Pero primero agreguemosle un README.md, un archivo de texto, o mejor dicho markdown, que se muestra por defecto si abrimos el repositorio en GitHub y se usa generalmente para explicar de qué se trata el proyecto y cómo usarlo (de ser software).



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner ^{*} Repository name ^{*}
 OTECPlatforma5Chile / Bootcamp 

Great repository names are short and memorable. Need inspiration? How about [solid-goggles?](#)

Description (optional)

- ☐  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☒  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: VisualStudio

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository

CHILE

Una vez que hayamos seteado nuestro repositorio remoto, pasemos a trabajar en el local, abramos la consola y creemos una nueva carpeta de trabajo, llámenla igual al repositorio que creamos en GitHub (no es obligatorio) e inicialicemos el repositorio.

```
florcita@florcita-OptiPlex-960:~/Escritorio/Bootcamp$ git init
Inicializado repositorio Git vacío en /home/florcita/Escritorio/Bootcamp/.git/
```

Ahora queremos relacionar este repositorio (vacío) al remoto en GitHub, para esto vamos a usar el comando `git remote`.

Remote

Veamos qué remotos tenemos asociados a este repositorio local.

```
$ git remote -v
```

Como no hubo output eso significa que no hay ninguna relación, así que creemos la nuestra. Usando `git remote add` vamos a poder hacer la vinculación.

Para eso en el botón color verde que dice CODE, desplegamos y nos aparece una URL para copiar, lo copiamos y lo pegamos en nuestra terminal.

```
$ git remote add origin https://github.com/OTECPlataforma5Chile/Bootcamp.git
```

Como podemos ver este comando tiene varios argumentos, `origin` es el nombre con el que vamos a asociar nuestro remoto, es el nombre por convención, pero podemos poner cualquier nombre que lo represente. Vamos a usar este nombre cuando nos conectemos con el servidor así que no se lo olviden! (aunque pueden buscarlo siempre en `git remote -v`).

El siguiente argumento es la dirección de el remoto, en mi caso es mi cuenta de github (OTECPlataforma5Chile), esto va a variar para cada uno de ustedes, ponga su usuario.

Y por último el nombre del repositorio que es *Bootcamp* en el ejemplo.

Para ver si el vínculo se produjo correctamente volvemos a correr

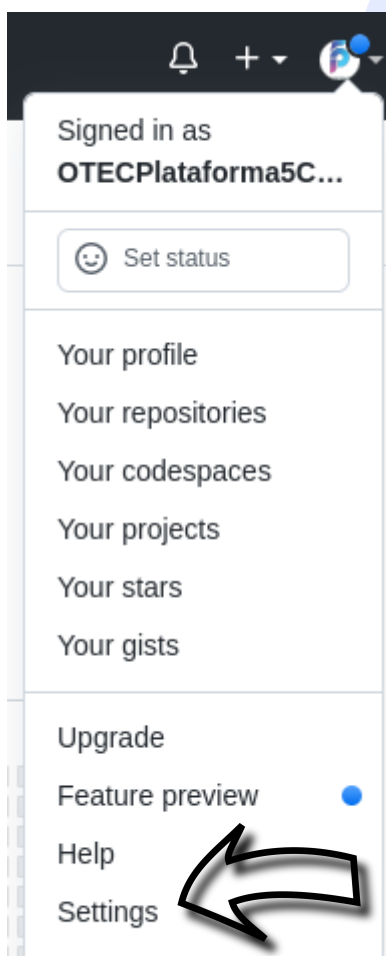
```
$git remote -v
```

```
florcita@florcita-OptiPlex-960:~/Escritorio/Bootcamp$ git remote -v
origin  https://github.com/OTECPlataforma5Chile/Bootcamp.git (fetch)
origin  https://github.com/OTECPlataforma5Chile/Bootcamp.git (push)
```

¡Genial, ya tenemos nuestros repositorios vinculados para ida y vuelta de información!

Generar un Token

Aquí dejamos un paso a paso para poder generar un token, necesario para poder hacer Pull, ya que nos piden la contraseña para pullear.



- Accessibility
- Notifications

Access

- Billing and plans
- Emails
- Password and authentication
- SSH and GPG keys
- Organizations
- Moderation

Code, planning, and automation

- Repositories
- Packages
- Pages
- Saved replies

Security

- Code security and analysis

Integrations

- Applications
- Scheduled reminders

Archives

- Security log
- Sponsorship log

- Developer settings

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can [@mention](#) other users and organizations to link to them.

URL

Twitter username

Company

You can [@mention](#) your company's GitHub organization to link it.

Location

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Contributions

Search or jump to... Pull requests Issues Marketplace Explore

Settings / Developer settings

GitHub Apps

- OAuth Apps
- Personal access tokens

GitHub Apps

New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

Personal access tokens

Generate new token

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).


Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Personal access tokens

[Generate new token](#)[Revoke all](#)

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_mvkJlecY3NnuigAzC5XNEQBGG5P0Zy4AHVb8 

[Delete](#)

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Este código token nos sirve para reemplazarlo por la contraseña necesaria.

Pull

Ahora ¿Cuál es la diferencia entre ambos repositorios? Que al local le falta el README.md. Cambiemos eso, vamos a usar el comando `git pull` pasando como argumento origin haciendo referencia al repositorio remoto (o el nombre que le hayan puesto) y luego `master` para especificarle que queremos la 'branch' principal del proyecto (todavía no se preocupe por qué significa eso).

```
$ git pull origin master
```

Si es la primera vez que usan git de manera remota va a pedir que nos identifiquemos, para poder hacerlo tenemos que configurar nuestra cuenta. Utilicen las mismas credenciales que GitHub.

Reemplace username por su usuario y mail por su mail registrado en GitHub.

```
$ git config --global user.name username
```

```
$ git config --global user.email mail
```

Finalmente deberíamos llegar a un output de este estilo.

```
$ git pull origin master
# remote: Counting objects: 3, done.
# remote: Total 3 (delta 0), reused 0 (delta 0),
pack-reused 0
```

```
# Unpacking objects: 100% (3/3), done.
# From
https://github.com/OTECPlataforma5Chile/Bootcamp.git
# * branch                master      -> FETCH_HEAD
# * [new branch]          master      -> origin/master
```

Una línea clave de este output es Unpacking objects: 100% (3/3), done., nos confirma que el pull salió como esperábamos y tenemos nuestro repositorio local igual al remoto (podemos comprobarlo en nuestra carpeta test).

Push

Así como podemos traer información de nuestro remoto también podemos enviar, y lo vamos a hacer con el comando git push.

Vamos a hacer un cambio en el README.md y después agregar un nuevo archivo. Asegurense de estar parados en la carpeta correcta.

```
$ echo '# Soy un titulo' > README.md // cambio el README (también
pueden usar su editor de texto)
```

```
$ touch app.js //creo el archivo
```

```
$ echo 'console.log("Hola")' > app.js // agrego un comando al archivo
(tambien pueden usar su editor de texto)
```

Si vemos el estado actual de nuestro repositorio, están los cambios.

```
$ git status
```

```
# On branch master
```

```
# Changes not staged for commit:
```

```
# (use "git add <file>..." to update what will be committed)
```

```
# (use "git checkout -- <file>..." to discard changes in working directory)
```

```
#
```

```
# modified: README.md
```

```
#
```

```
# Untracked files:
```

```
# (use "git add <file>..." to include in what will be committed)
```

```
#
```

```
# app.js
```

#

no changes added to commit (use "git add" and/or "git commit -a")

Ahora tenemos un archivo no guardado y uno no 'trackeado', guardemos los cambios como sabemos hacerlo.

```
$ git add *
```

```
$ git commit -m 'cambio README y agrego app.js'
```

```
# [master 99187e6] cambio README y agrego app.js
```

```
# 2 files changed, 2 insertions(+), 1 deletion(-)
```

```
# create mode 100644 app.js
```

Ahora nuestro repositorio local está guardado con la última versión disponible, pero si vemos el repositorio remoto de GitHub sigue con la versión anterior. Esto se debe a que son 2 repositorios aislados, que si bien están vinculados entre ellos, no significa que los cambios se hagan automáticamente.

Vamos a copiar nuestro repositorio local hacia el remoto. Usamos el comando `git push` y como argumento pasamos el nombre que le pusimos al remoto (en mi caso 'origin') y luego la branch que estamos trabajando (por defecto 'master').

```
$ git push origin master
```

Nos va a pedir username y password para ver si tenemos permitido modificar este repositorio.

```
$ git push origin master
```

```
# Username for 'https://github.com': OTECPlataforma5Chile
```

```
# Password for 'https://OTECPlataforma5Chile@github.com':
```

```
# Counting objects: 4, done.
```

```
# Delta compression using up to 4 threads.
```

```
# Compressing objects: 100% (2/2), done.
```

```
# Writing objects: 100% (4/4), 345 bytes | 0 bytes/s, done.
```

```
# Total 4 (delta 0), reused 0 (delta 0)
```

```
# To https://github.com/facuveelasco/test
```

```
# 6d51e91..99187e6 master -> master
```


Si ahora pasamos a GitHub en el browser, debería haber actualizado los cambios y los repositorios se deberían ver iguales.

Clone

Nos va a pasar que no siempre empezamos nosotros un repositorio desde cero, muchas veces queremos arrancar con un repositorio ya existente, para eso podemos usar git clone. Con este comando vamos a poder generar automáticamente un repositorio local completamente igual al remoto y ya vinculados entre sí.

Entonces salgamos de la carpeta test y borremosla.

```
$ cd .. // el doble punto indica la carpeta superior/anterior
```

```
$ rm -rf test
```

Y ahora clono el repositorio que esta en GitHub ya armado.

```
$ git clone https://github.com/facuvelasco/test
```

```
# Cloning into 'test'...
```

```
# remote: Counting objects: 7, done.
```

```
# remote: Compressing objects: 100% (3/3), done.
```

```
# remote: Total 7 (delta 0), reused 4 (delta 0), pack-reused 0
```

```
# Unpacking objects: 100% (7/7), done.
```

```
# Checking connectivity... done.
```

Ahora se nos creó la carpeta test con los archivos que subimos antes, ¡pruébenlo!

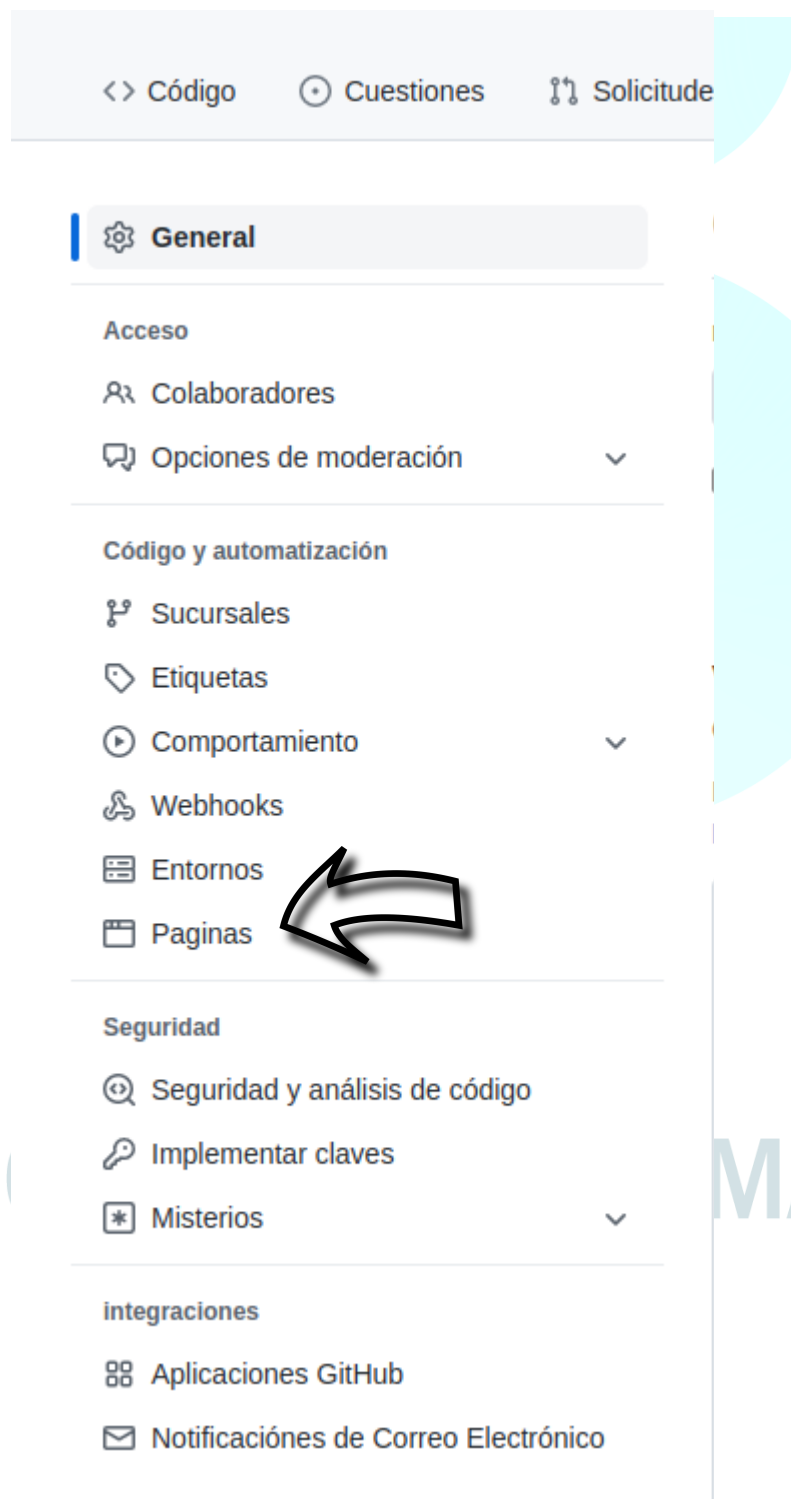
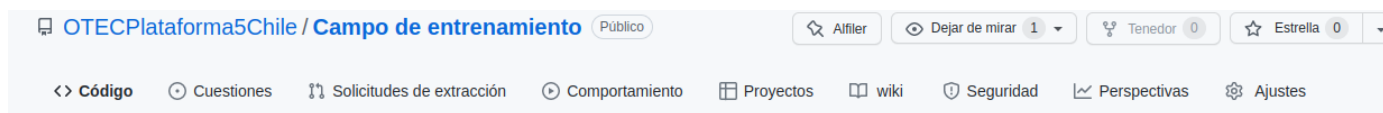
```
$ cd test
```

```
$ ls
```

```
# app.js README.md
```

Con esos comandos básicos ya podemos guardar las diferentes versiones de nuestro proyecto en un servidor remoto.

¿Cómo obtener el link de nuestro repertorio?



MA 5 CHILE

Páginas de GitHub

GitHub Pages está diseñado para alojar sus páginas personales, de organización o de proyecto desde un repositorio de GitHub.

Fuente

Las páginas de GitHub están actualmente deshabilitadas. Seleccione una fuente a continuación para habilitar las páginas de GitHub para este repositorio. [Más información](#)

🔗 Rama: principal ▾

📁 / (raíz) ▾

Salvar



Selector de temas

Seleccione un tema para publicar su sitio con un tema de Jekyll usando la gh-pagesrama. [Más información](#)

Elige un tema

Páginas de GitHub

GitHub Pages está diseñado para alojar sus páginas personales, de organización o de proyecto desde un repositorio de GitHub.

📘 Tu sitio está listo para ser publicado en <https://otecplataforma5chile.github.io/Bootcamp/>



OTEC PLATAFORMA 5 CHILE