



Terminal, Git, GitHub y GitHub Pages

Introducción a GitHub

Crear un repositorio remoto en Github para controlar las versiones de un proyecto y publicar la página web utilizando Github Pages.

- Unidad 1: HTML y CSS
- Unidad 2: Bootstrap
- Unidad 3: JavaScript
- Unidad 4: Terminal, Git, GitHub y GitHub Pages



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Aplica el procedimiento de subida del código versionado mediante una conexión SSH, para la mantención de un repositorio remoto.*

`/* Introducción a GitHub */`

Introducción a GitHub

¿Qué es GitHub?

- Existen varios tipos de repositorios remotos y empresas asociadas a proporcionarlos, las más usadas son: GitHub, Bitbucket y Gitlab.
- GitHub es un gestor de repositorios remotos, lo que quiere decir que podemos almacenar una copia de nuestro código en sus servidores. Así podemos trabajar colaborativamente y respaldar nuestro trabajo.
- GitHub es gratis y no tiene restricciones de la cantidad de repositorios que podamos crear.

Introducción a GitHub

Configuración de GitHub

- Existen varias formas de hacerlo, pero la más fácil y segura es utilizar el protocolo SSH y sus llaves. Las llaves, son un medio por el cual podemos identificar nuestro equipo con un servidor o página específica, incluso sin tener que ingresar una contraseña. Esto funciona mediante dos llaves, una privada y otra pública. La privada vive en nuestro equipo y la pública es la que ingresa en el lado remoto.
- Si quieres conocer el procedimiento para conectar tu cuenta de Github con SSH, te invitamos a leer la [documentación oficial](#).

/* Subiendo y bajando cambios */

Subiendo y bajando cambios

- Para subir los cambios al repositorio remoto debemos utilizar el comando:

```
git push origin main
```

- De esta forma se suben todos los cambios registrados (comiteados) al repositorio remoto, en la branch (rama) en la que me encuentre, en este caso: **main**.

Subiendo y bajando cambios

- Ahora, si por el contrario necesito bajar los cambios que están en un remoto que tenemos registrado en nuestro proyecto, podremos utilizar el comando:

```
git pull origin main
```

- Lo cual traerá a nuestro computador todos los cambios que se hayan realizado en el remoto, uniendo de forma automática los archivos que encuentre con cambios.
- Recuerda que el nombre **origin** corresponderá al remoto registrado en tu proyecto y main se refiere a la rama main del remoto.

/* Manejo de repositorios remotos */

Manejo de repositorios remotos

git remote

- Si necesitamos saber si el proyecto en que estamos trabajando ya contiene alguna referencia a un repositorio remoto, lo realizaremos con el comando:

git remote

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote
origin
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ |
```

Manejo de repositorios remotos

git remote -v

- Si necesitamos saber las url de estos servidores usamos el comando:

```
git remote -v
```

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote -v
origin  https://github.com/alegonzalezcelis/Meet-Coffee.git (fetch)
origin  https://github.com/alegonzalezcelis/Meet-Coffee.git (push)

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ |
```

Manejo de repositorios remotos

Añadiendo un repositorio remoto

- Para añadir un repositorio remoto, simplemente debemos usar el comando:

```
git remote add [nombre] [dirección del repositorio]
```

- Si este comando se ejecuta bien, no muestra ningún resultado, solo debemos corroborarlo con **git remote** o **git remote -v**.
- Generalmente, al añadir un repositorio remoto se utiliza el nombre "origin". Sin embargo, podemos tener más de un servidor remoto configurado, solo tendríamos que utilizar nombres distintos.

Manejo de repositorios remotos

Obteniendo información de un repositorio remoto

- Una vez que ya tengamos añadido un repositorio remoto en nuestro proyecto, podremos obtener información de él con el siguiente comando (cambiando el nombre por el de nuestro repositorio):

```
git remote show [nombre]
```

```
Ale@DESKTOP-6FQK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/alegonzalezcelis/Meet-Coffee.git
  Push URL: https://github.com/alegonzalezcelis/Meet-Coffee.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
```

/* Manejo de repositorios remotos */

Manejo de repositorios remotos

git remote rename

- Si necesitamos renombrar un repositorio remoto que hemos añadido, podemos realizarlo de la siguiente forma:

```
git remote rename nombreActual NuevoNombre
```

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote rename origin origin_new

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote
origin_new
```


Manejo de repositorios remotos

git remote rm

- Si necesitamos borrar un repositorio remoto, podemos realizarlo con el siguiente comando git:

```
git remote rm NombreRepositorio
```

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote rm origin_new

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote -v

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ |
```

***/* Repositorios que ya están
en GitHub */***

Repositorios que ya están en GitHub

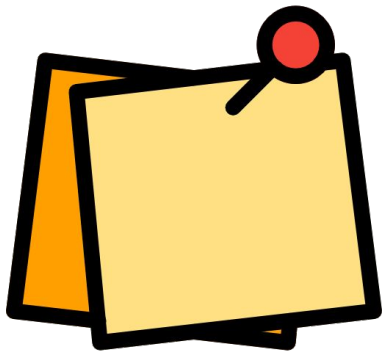
git clone

- GitHub nos proporciona herramientas para utilizar repositorios que ya están creados. Para ellos existen dos herramientas que nos ayudarán: Fork y Clone.
- La primera es Fork, este comando creará una bifurcación del repositorio, en otras palabras realizará una copia del proyecto completo en nuestra propia cuenta de GitHub.
- El siguiente paso para comenzar a trabajar sería bajar el contenido de este repositorio hacia nuestro computador local. Esto lo realizaremos clonando el repositorio.

```
git clone [dirección del repositorio]
```

Introducción a GitHub

Importante



GitHub en sus inicios definió la rama master como su rama principal o por defecto, sin embargo, desde octubre de 2020, han implementado cambios en su rama principal, sustituyendo gradualmente la rama master por la nueva rama main, este cambio afecta directamente a los nuevos repositorios. Según se puede apreciar en su documentación oficial, el cambio responde a la implementación de un nombre más corto y a la interpretación más óptima en otros lenguajes. Para más información acerca de esta actualización visita el siguiente link:

<https://github.com/github/renaming>

Resumen

- GitHub es un gestor de repositorios remotos, lo que quiere decir que podemos almacenar una copia de nuestro código en sus servidores. Así podemos trabajar colaborativamente y respaldar nuestro trabajo.
- Las llaves, son un medio por el cual podemos identificar nuestro equipo con un servidor o página específica, incluso sin tener que ingresar una contraseña.
- Para subir los cambios al repositorio remoto debemos utilizar el comando `git push origin main`.
- Si necesitamos saber si el proyecto en que estamos trabajando ya contiene alguna referencia a un repositorio remoto, lo realizaremos con el comando `git remote`, el cual nos mostrará el nombre de los repositorios que tengamos añadidos.
- Fork es un comando que creará una bifurcación del repositorio, en otras palabras realizará una copia del proyecto completo en nuestra propia cuenta de GitHub.

¿Existe algún concepto que
no hayas comprendido?





Próxima sesión...

- *Aplica el procedimiento de habilitación de una página web con Github Pages, para que el contenido esté disponible públicamente.*

{desafío}
latam_

*Academia de
talentos digitales*

