

Programación para la Ciencia de Datos con R y Python

Clase 1: Github

2024-09-10

- Github contribuye a solucionar el problema del “Documento_Final.docx”.
- Github trackea todos los cambios hechos a un documento (guarda todos los deltas con timestamps).
- Github también soluciona:
 - ▶ Problemas de copia conflictiva (Dropbox, etc).
 - ▶ No puedo reproducir los resultados de mi código.
 - ▶ Quién escribió esto hace muchos años (Autoría).

3 conceptos claves:

- Clonar
- Commit
- Branch (rama)

- Clonar es similar a descargar un repositorio.
- La diferencia entre clonar y descargar es que **Github recuerda de dónde descargamos el repositorio**. Necesario para saber a dónde mandar nuestros cambios cuando seamos contribuidores.

Cómo clonar un repo

- 1 Ve a tu proyecto favorito en www.github.com
- 2 Haz click en el botón verde que dice *Code*
- 3 Dale click en *Open with Github Desktop*
- 4 Selecciona la carpeta de tu computadora donde se clonará el repo.
NO lo clones en una carpeta compartida de Dropbox, Box, etc.

- Los commits son la manera como github maneja el control de versiones.
- Para no tener que guardar el estado completo de un documento en tal o cual versión, Git **usa los commits para indicar las diferencias significativas entre versión y versión**
- Cada commit es una captura de los archivos del proyecto en determinado momento, y lo compara respecto al commit anterior.
- Cada commit tiene un timestamp (dentro de este ya tiene un id que incluye fecha y hora)

Pequeño detour: Creemos un repo

- Github servirá para almacenar, administrar y seguir proyectos individuales.
- 1 Ve a la página principal de Github. Dale click al botón verde que dice *New*

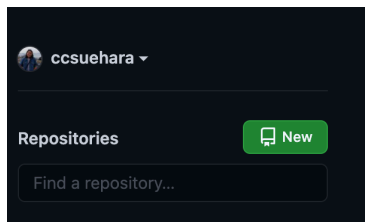


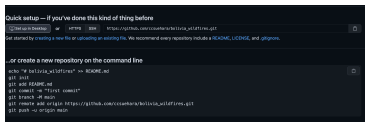
Figure 1: “ ”

Pequeño detour: Creemos un repo

- 2 Luego, elige el nombre del repo. También tenemos otras opciones, como agregar una descripción del repo, volverlo público/privado, agregarle un `.gitignore`, un `README.md`, y una licencia. Creen el repositorio!

Pequeño detour: Creemos un repo

- 3 Una vez creado el repo, te dan las siguientes instrucciones. Dale click a *Set up in Desktop*



```
Quick setup — If you've done this kind of thing before
Set started by creating a new file or validating an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line
$ git init
$ git add README.md
$ git commit -m "First commit"
$ git branch -M main
$ git remote add origin https://github.com/inec/inec.git
$ git push -u origin main
```

Figure 2: “ ”

Pequeño detour: Creemos un repo

- 4 Aparecerá el siguiente cuadro, acepten

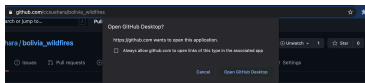


Figure 3: “ ”

- 5 Finalmente, github les pedirá que elijan la carpeta asociada al repo. Elijan una carpeta que **no esté en Dropbox, Box, etc**

- Ahora que tenemos un repo creado, hagamos un commit
- Hecho!

3. Branch

- Los branches **permiten crear una copia del código donde puedes experimentar/jugar/desarrollar** con él. Si te gusta el resultado, **puedes mergear el experimento a la versión principal.**
- Esta versión no-lineal es más fidedigna a cómo se trabaja en la vida real.

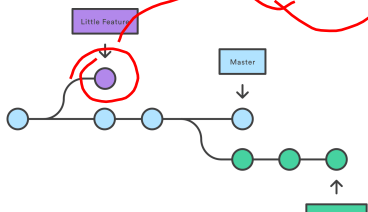


Figure 4: “ ”

Muchas gracias.



www.inei.gob.pe/enei