

Version Control

Control de versiones con Git y RStudio



R-Ladies General Pico
R-Ladies Buenos Aires
R-Ladies Santa Rosa



¿Quién sos vos?

Romina trabaja ordenando y analizando datos utilizando R para una variedad de clientes.

Utiliza proyectos en RStudio para ordenar su trabajo.

Comparte sus avances y resultados utilizando herramientas en la nube (como dropbox y google drive).

Compartir de esta manera le ha traído varios dolores de cabeza

Sabe que Git puede ayudarla con estos problemas pero no le queda claro como.

Tiene usuario en GitHub pero nunca usó.

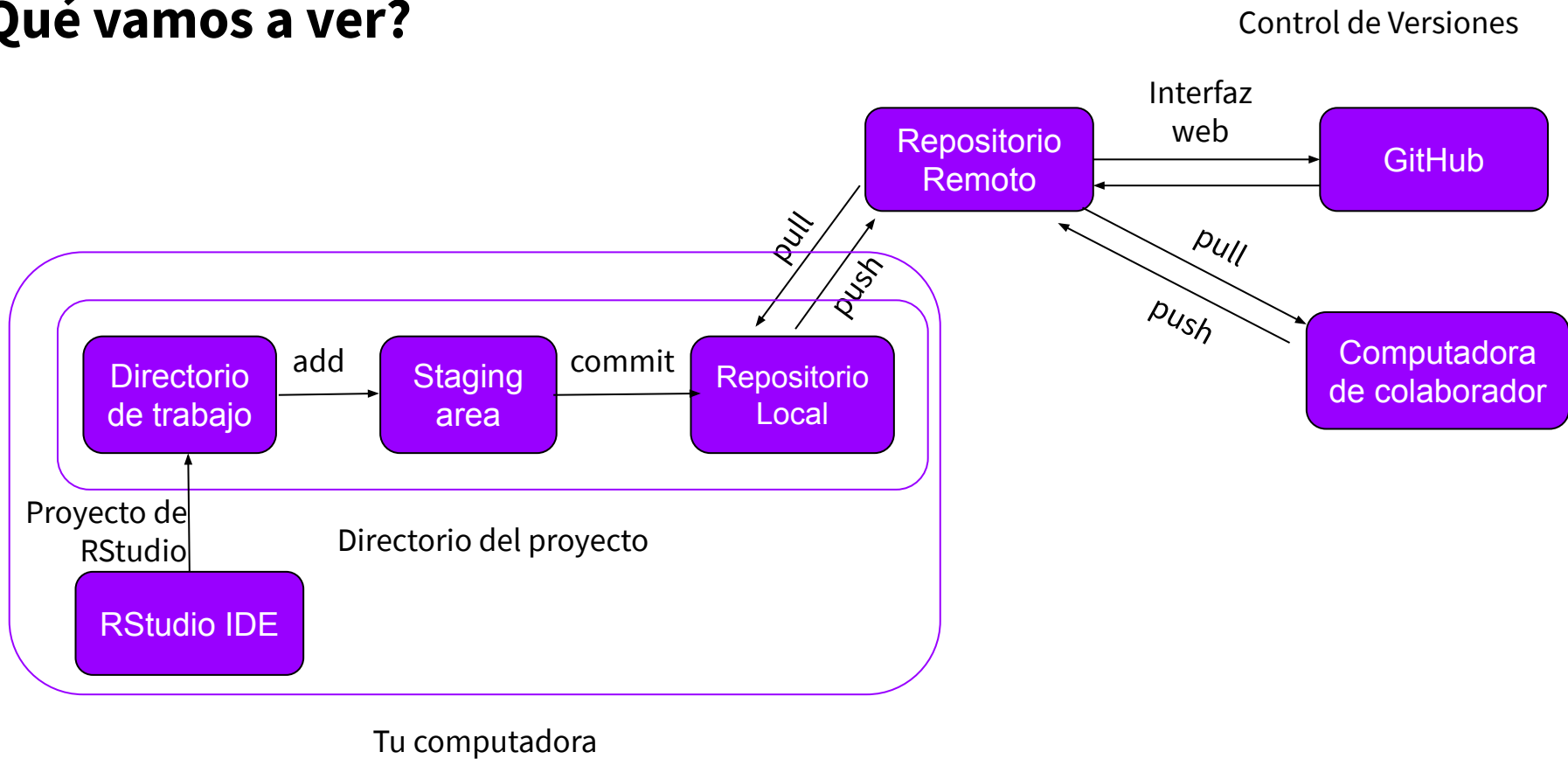
Quiere entender como funciona y como usarlo con R y RStudio para poder incorporarlo a sus proyectos.



Primer paso: ¿Qué es versionar código?

- El versionado almacena todas las modificaciones realizadas en el código.
- Permite acceder a versiones anteriores de cualquier archivo.
- Garantiza el trabajo en equipo de manera eficiente.
- Acciones útiles: regresar a una versión anterior de tu proyecto, comparar cambios, ver quien realizó y para que una modificación, recuperar archivos perdidos.... Y MUCHO MAS !

¿Qué vamos a ver?



¿Cómo empezamos?

Opciones:

- Creamos un repositorio local y luego lo publicamos en el servidor remoto
- Creamos el repositorio en el servidor remoto y luego lo descargamos en nuestro directorio local.
- Bajamos un repositorio existente desde el servidor remoto, mediante el comando : `git clone <remote>`

Comandos básicos: Add

Sinopsis : *git add*

Descripción:

Agrega el archivo o directorio al Staging Area.

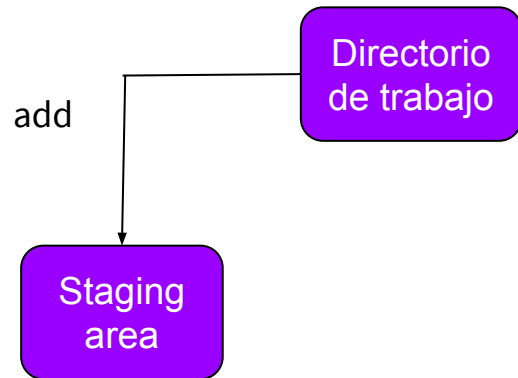
Opciones:

```
git add <file-path>
```

```
git add -all
```

```
git add -u
```

```
git add .
```



Comandos básicos: Commit

Sinopsis: *git commit*

Descripción:

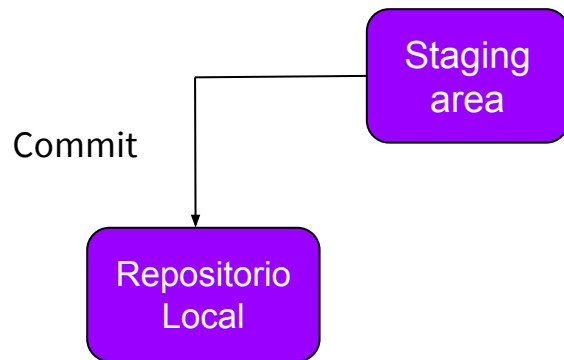
Agrega/guarda los cambios al repositorio local.

Opciones:

```
git commit -m "Primer commit"
```

```
git commit -am "Primer commit"
```

```
git revert "hash_commit"
```



Comandos básicos: Pull

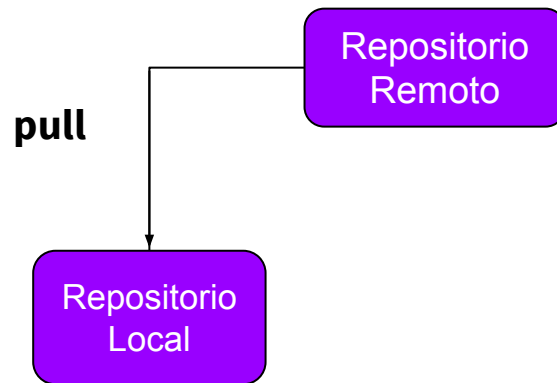
Sinopsis: *git pull*

Descripción:

Actualiza el repositorio local con el repositorio remoto.

Opciones:

```
git pull <remote>
```

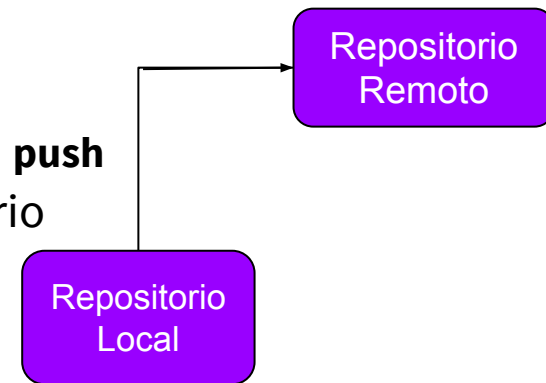


Comandos básicos: Push

Sinopsis: *git push*

Descripción:

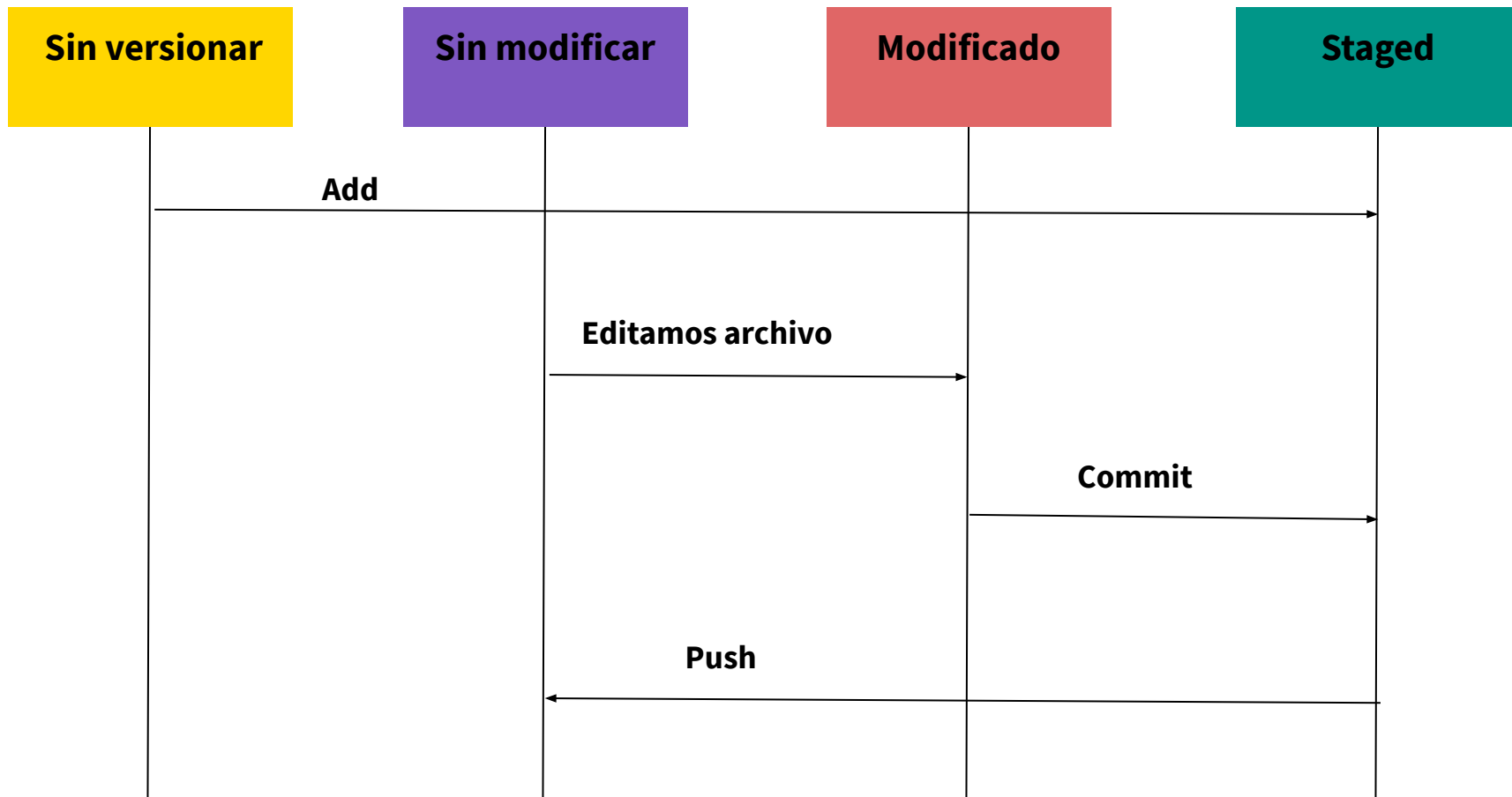
Envía los cambios desde el repositorio local al repositorio remoto



Opciones:

`git push`

`git push -- force`



Ciclo de vida de los archivos mediante Git

Para tener en cuenta :)

- Ejecutar pull antes de hacer un push.
- Quedarnos tranquilxs que Git gestiona los conflictos.
- Siempre se puede volver a una versión anterior, solucionar, romper y volver a solucionar :)

¿Qué software necesitamos?



**Git
(consola)**



Desktop



Studio[®]

**IDEs para
manejar
Git**



**Git en la
web**

Nuevo proyecto, GitHub primero

- Crear repositorio en Github
- Crear proyecto en RStudio relacionandolo con el repo de Github
- Usar control de versiones!




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 *



 yabellini ▾

 / PruebaMeetup 

Great repository names are short and memorable. Need inspiration? How about **urban-octo-train**?

Description (optional)

Prueba repo meetup ladies

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

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

- ☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

 |

Add a license: **None** ▾



Create repository

yabellini / PruebaMeetup

Unwatch

1

★ Star

0

🔗 Fork

0

<> Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security 0

Insights

Settings

Prueba repo meetup ladies

Edit

Manage topics

1 commit

1 branch

0 packages

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



yabellini Initial commit



README.md

Initial commit



README.md

PruebaMeetup

Prueba repo meetup ladies

Clone with HTTPS

Use SSH

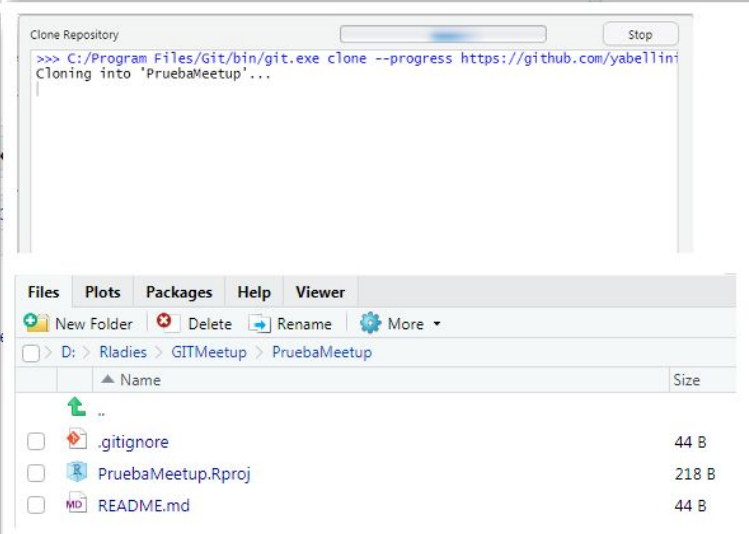
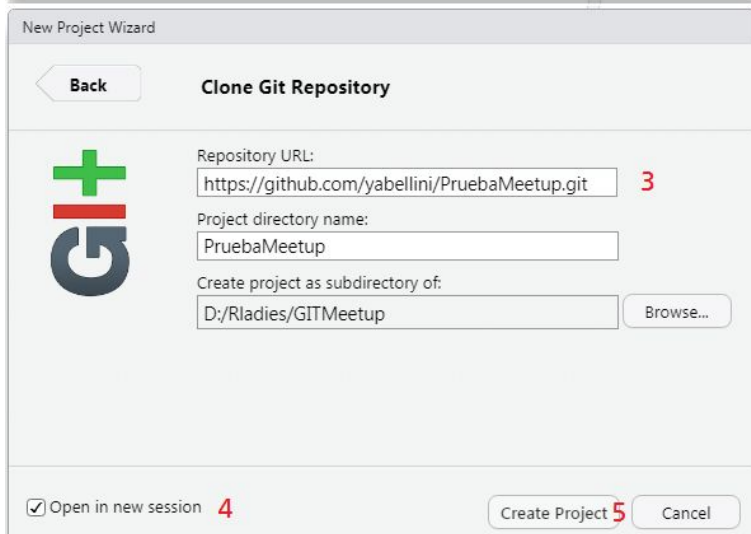
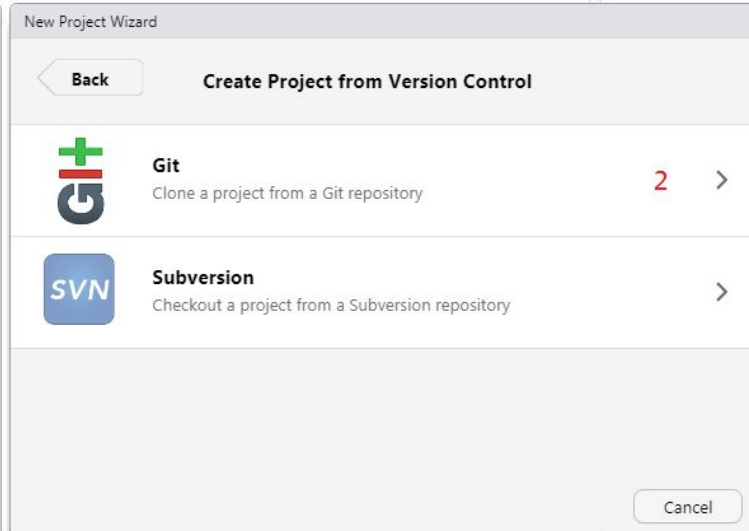
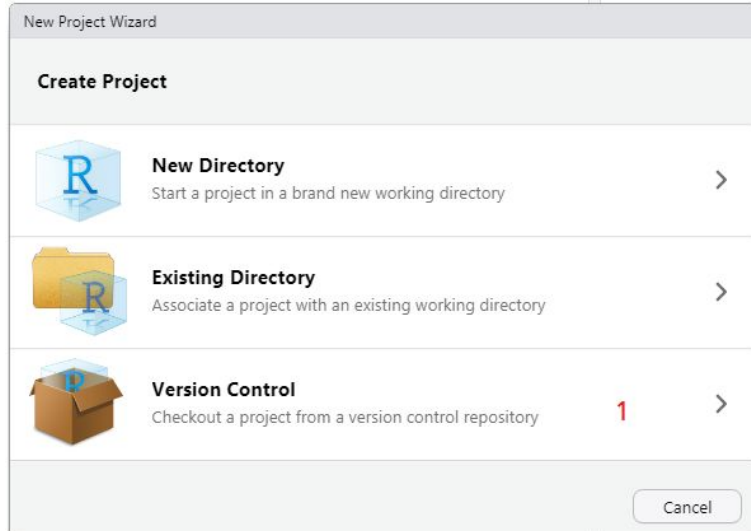
Use Git or checkout with SVN using the web URL.

https://github.com/yabellini/PruebaMeetup



Open in Desktop

Download ZIP



Proyecto existente, GitHub primero



- Crear repositorio en Github
- Crear proyecto en RStudio relacionandolo con el repo de Github
- Copiar en este proyecto todos los script, datos, etc.
- Usar control de versiones!

Usando control de versiones



- Modificar en RStudio y enviar al repo remoto.
- Modificar en el repo remoto y bajar los cambios a RStudio.

¿ Cómo seguimos ?

- Post con todo el material del meetup:
<https://yabellini.netlify.app/es/post/githubconr/>
- Happy Git and GitHub for the useR:
<https://happygitwithr.com/index.html>





<https://latin-r.com/>

31 de Mayo
Presentación de
trabajos

Somos MetaDocencia

- <https://metadocencia.netlify.com>
- <https://github.com/metadocencia>
- metadocencia@gmail.com
- <https://metadocencia.slack.com>
- <http://tiny.cc/youtubeMetaDocencia>
- Twitter: @metadocencia



Vamos a ver tus preguntas en el chat



@yabellini @marysol_gatti
@rladiesSR @rladies_pico @rladiesBA