

Git y Github

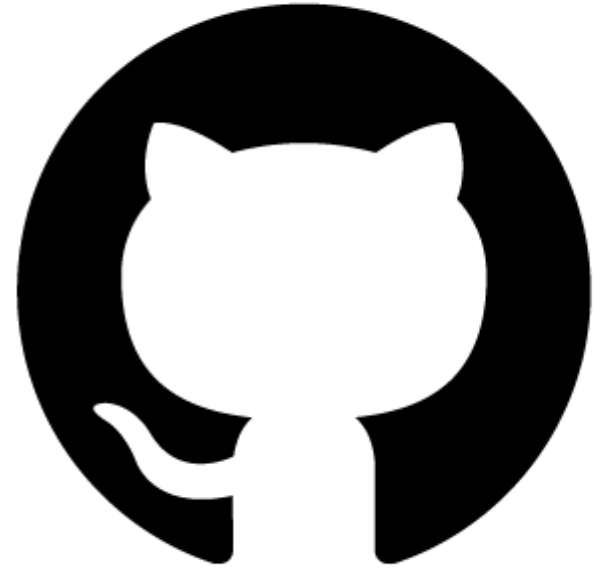
Ph D. Stephanie Hereira Pacheco

CTBC-UATx

2022-09-07



git



GitHub

Git y versiones de control

Para la clase de hoy vamos a necesitar:

- R y Rstudio (últimas versiones recomendable)
- Tener una cuenta Github
- Tener instalado Git:

Instala **Git** de Aquí.

Git y versiones de control

- Git es un programa o sistema de control de versiones gratis y de código abierto.
- Fue creado por Linus Torvalds en 2005.
- Su propósito original era ayudar a grupos de desarrolladores a trabajar en colaboración en grandes proyectos de software.
- Git administra la evolución de un conjunto de archivos, llamado repositorio, de una manera sensata y altamente estructurada (control de cambios).

Git y versiones de control

- Los proyectos cambian a lo largo del tiempo y colaborar puede ser complicado con códigos. Además puedes almacenar tus códigos en un repositorio.
- Git funciona con GitHub, Bitbucked o Gitlab
- Con Git cada colaborador tiene una copia del repositorio central en su pc, con todos los archivos y la historia de los cambios por los que han pasado.

Github

Github

- GitHub es una plataforma o servicio web que nos permite guardar y compartir repositorios de Git que podemos usar como servidores remotos.
- Después de crear nuestra cuenta, podemos crear o importar repositorios, crear organizaciones y proyectos de trabajo, descubrir repositorios de otras personas, contribuir a esos proyectos, dar estrellas y muchas otras cosas.
- Muchos de los programas que conocemos y usamos están en repositorios en Github en incluso con el paquete devtools de R podemos instalarlos.

Ejemplo:

```
if (!requireNamespace("devtools", quietly = TRUE)){install.packages("devtools")}  
devtools::install_github("jbisanz/qiime2R")
```


Git y Github

Otras ventajas...

- Facilita la contribución a proyectos de código abierto
- Opciones de integración: Git y Github pueden integrarse con otras plataformas como Amazon y Google cloud
- Te permite rastrear los cambios a través de versiones (filogenia/GoogleDocs)
- Te permite compartir y mostrar tu trabajo
- La comunidad de R es muy activa en GitHub. Hay muchos paquetes ahí, como el tidyverse. GitHub y RStudio se conectan muy bien.

Diferencias entre Git y Github

GIT	GitHUB
Es un programa	Es un servicio
Se instala localmente	Está en la web
Herramienta para gestionar versiones	Espacio para subir una copia del repositorio

¡ Empecemos !

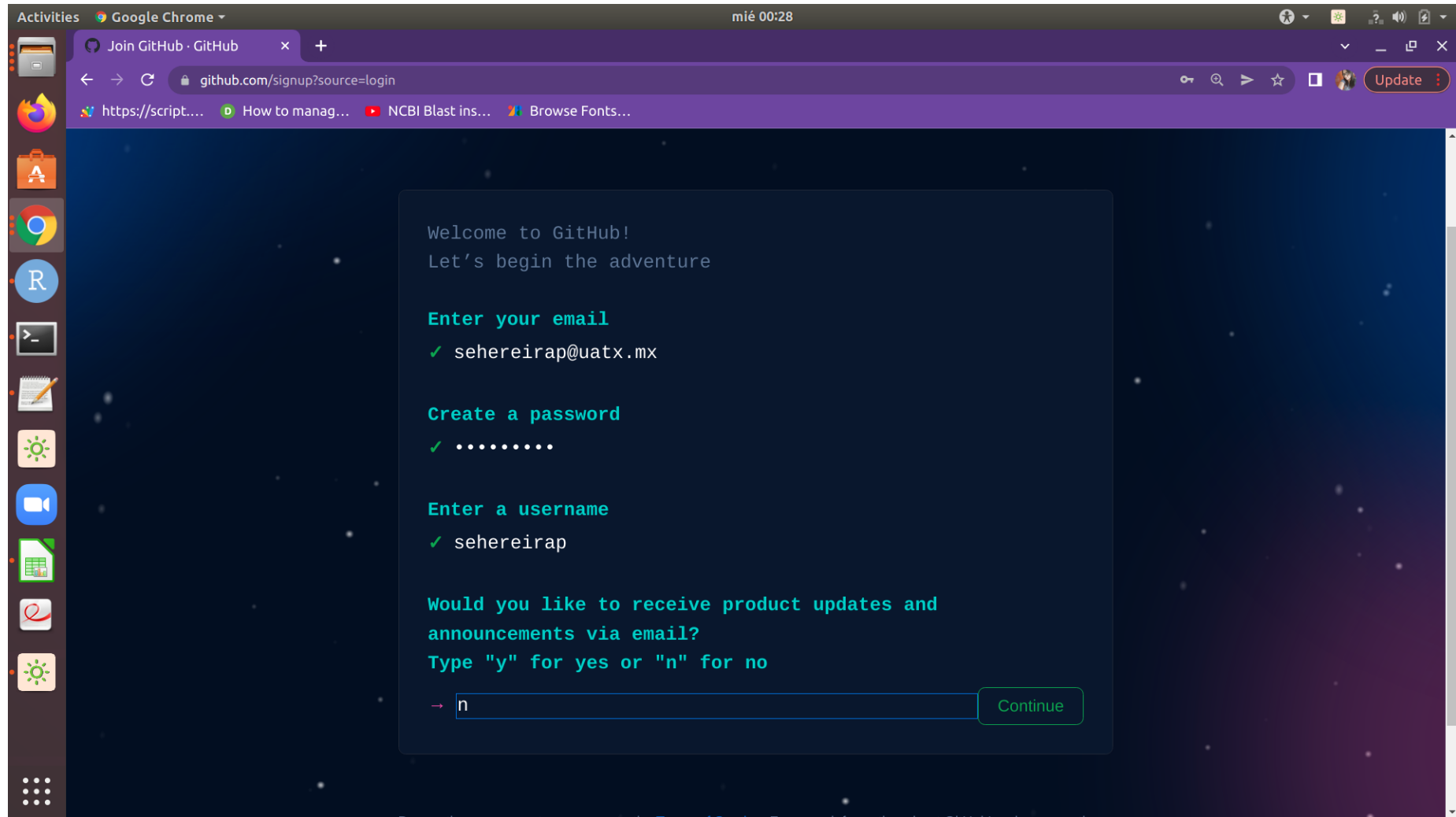
Antes . . .

1. Tener instalado Git
2. Tener cuenta en GitHub
3. Tener instalado R y Rstudio
4. Git y Github se conecten
5. R esté enlazado con el repositorio (Tener un Proyecto de R para cada trabajo o repositorio)

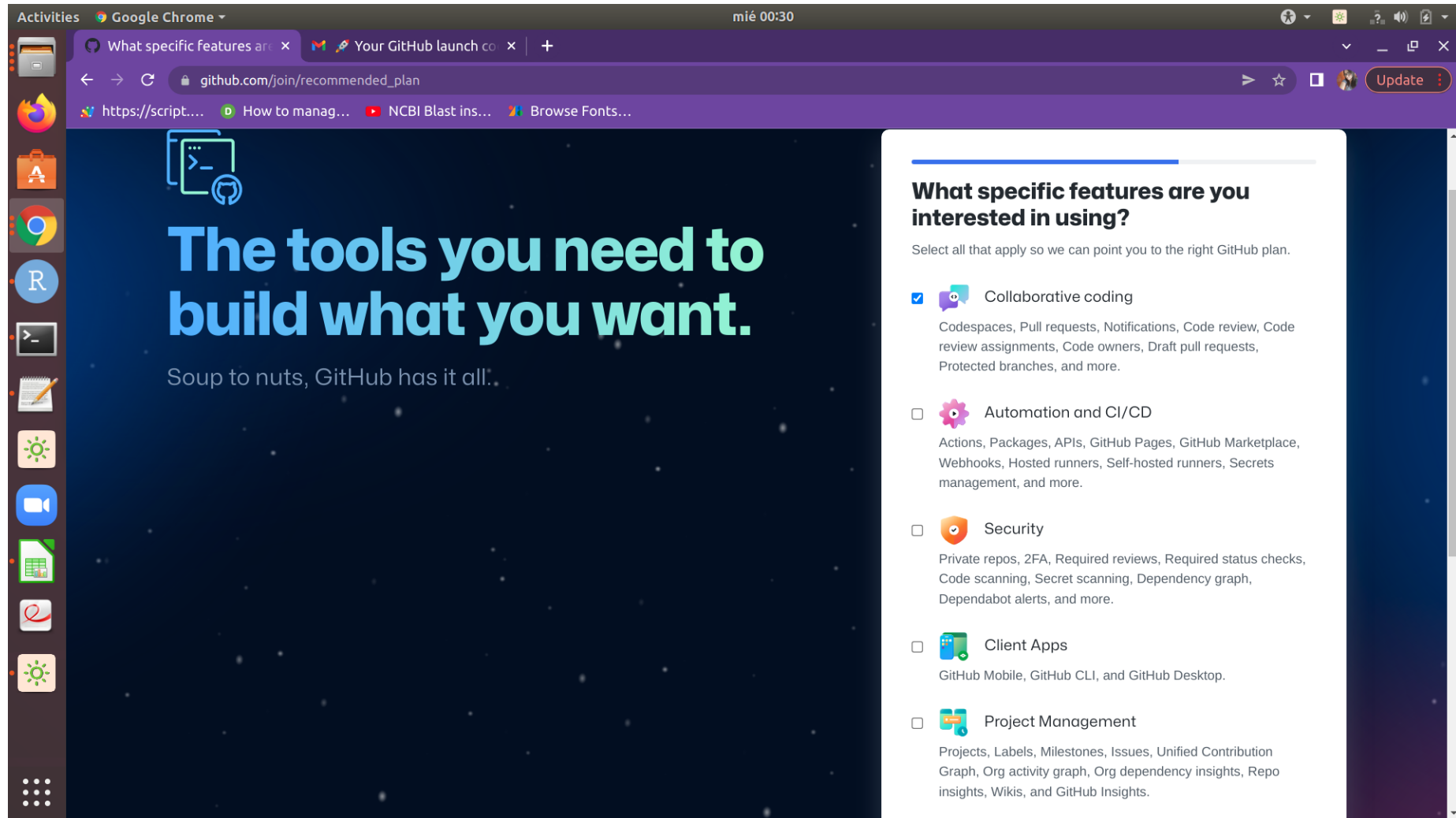
Checando si tenemos Git instalado

```
which git y git --version
```

Cuenta en Github.



Qué harás?....



Tú perfil...

Activities Google Chrome mié 00:31

sehereirap x Your GitHub launch co x +

github.com/sehereirap

https://script... How to manag... NCBI Blast ins... Browse Fonts...

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

Overview Repositories Projects Packages Stars

Popular repositories Customize your pins

You don't have any public repositories yet.

sehereirap

Edit profile

Joined 2 minutes ago

1 contribution in the last year Contribution settings

	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
Mon												
Wed												
Fri												

Learn how we count contributions Less More

This is your **contribution graph**. Your first square is for joining GitHub and you'll earn more as you make [additional contributions](#). More contributions means a darker green square for that day. Over time, your chart might start looking [something like this](#).

We have a quick guide that will show you how to create your first repository and earn more green squares!

[Read the Hello World guide](#)

Contribution activity

Creando un repositorio

- Entrar a "Your repositories" y darle crear repositorio nuevo o "New", lo nombramos como queramos.
- Añadir una descripción e iniciar el repositorio con un README.
- Lo demás en default.
- Dar click en "Create repository". Hagámoslo juntos.

Crear Token en GitHub

1. En tu perfil de GitHub, dale click a la imagen de perfil (en la esquina superior derecha), y busca la opción de Settings o configuración.
2. Da click a continuación en Developer settings ó Parámetros del desarrollador (hasta abajo).
3. En la barra lateral izquierda da click en Tokens de acceso personal.
4. Haz click en Generar un nuevo token.
5. Escoger el tipo de token dependiendo de lo que se hará (repo, workflow, etc) y el tiempo del token. Y da Generar token.
6. Sale el token y debes guardarlo en un txt o correo o en algún lugar para no perderlo (Sólo lo muestra una vez). Hagámoslo juntos.

Configurar las credenciales

- Algunos sistemas operativos o pcs guardan el username y password, así que no lo tenemos que poner en cada push en otras se pone una sola vez.
- Antes te pedían siempre el user y password pero ahora por nuevas políticas piden el token.

Para configurarlo por primera vez, esto lo hacemos desde la terminal:

```
git config --global user.name "sehereirap"  
git config --global user.email sehereirap@uaxt.mx  
git config --global --list
```

Probemos nuestro Git en terminal

Clonemos un repositorio para ejemplo

```
git clone https://github.com/Steph0522/R_Basics_Lab53.git
cd R_Basics_Lab53/
ls -la
head README.md
```

Hagamos cambios en este repo

```
echo "Pruebo agregar esto" >> README.md
git status
git add README.md
git commit -m "A commit from my local computer"
git pull
git push
```

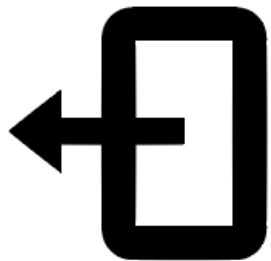
In case of fire



1. `git commit`



2. `git push`



3. leave building

Algunas recomendaciones...

- Siempre usar git pull antes de git push ó podrán haber errores de versiones.
- Usar mensajes INFORMATIVOS en los commits.
- Verifica si los cambios se subieron a GitHub



Conectado Git y Github con Rstudio

Vamos a Rstudio y vamos configurar allí credenciales y PAT (token) con el paquete **usethis**:

```
usethis::use_git_config(user.name = "sehereirap", user.email = "sehereirap@uatx.mx")  
usethis::edit_r_environ()
```

Les va a salir una línea donde pondrán su token, algo así, cuidar espacios:

```
GITHUB_PAT=ghp_KtgbrihBJq5BaqdN6aaacQgtXuaEDkrN2eLSZ7
```

Dependiendo del sistema operativo y pc ya en Rstudio no les pedirá más su user o token. En algunos casos sí lo seguirá pidiendo.

Conectado Git y Github con Rstudio

Ahora sí vamos cargar o clonar nuestro repositorio en Rstudio:

1. Crea un proyecto en RStudio: File > New Project > Version Control > Git. Ahi pega el URL del repositorio.
2. Git va a aparecer en la esquina superior.

Esto que acabamos de hacer nos permitirá tener un nuevo directorio, proyecto y el repositorio de GitHub enlazado a nuestro Rstudio.

Commit, pull y push en RStudio

1. En la parte superior de nuestro Rstudio vamos a ver **Git**
2. Vamos a hacer algún cambio en nuestro README.md
3. Damos click en commit.
4. Escribimos un mensaje útil
5. Luego damos Push

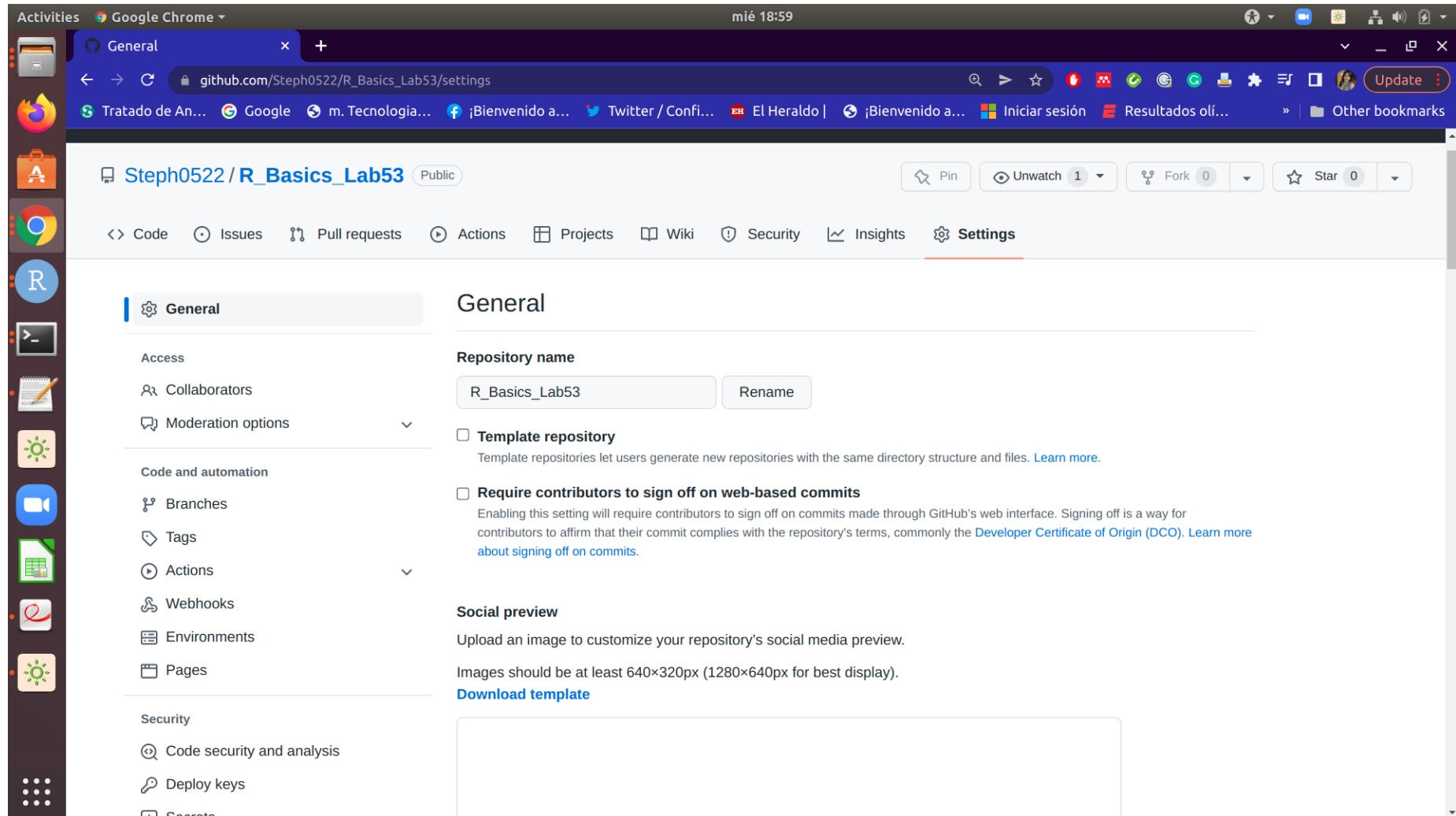
Nota: otra forma que tenemos para clonar un repo es descargar el zip.
Recueden cuando trabajan en colaboración siempre hacer pull.

Proyectos colaborativos

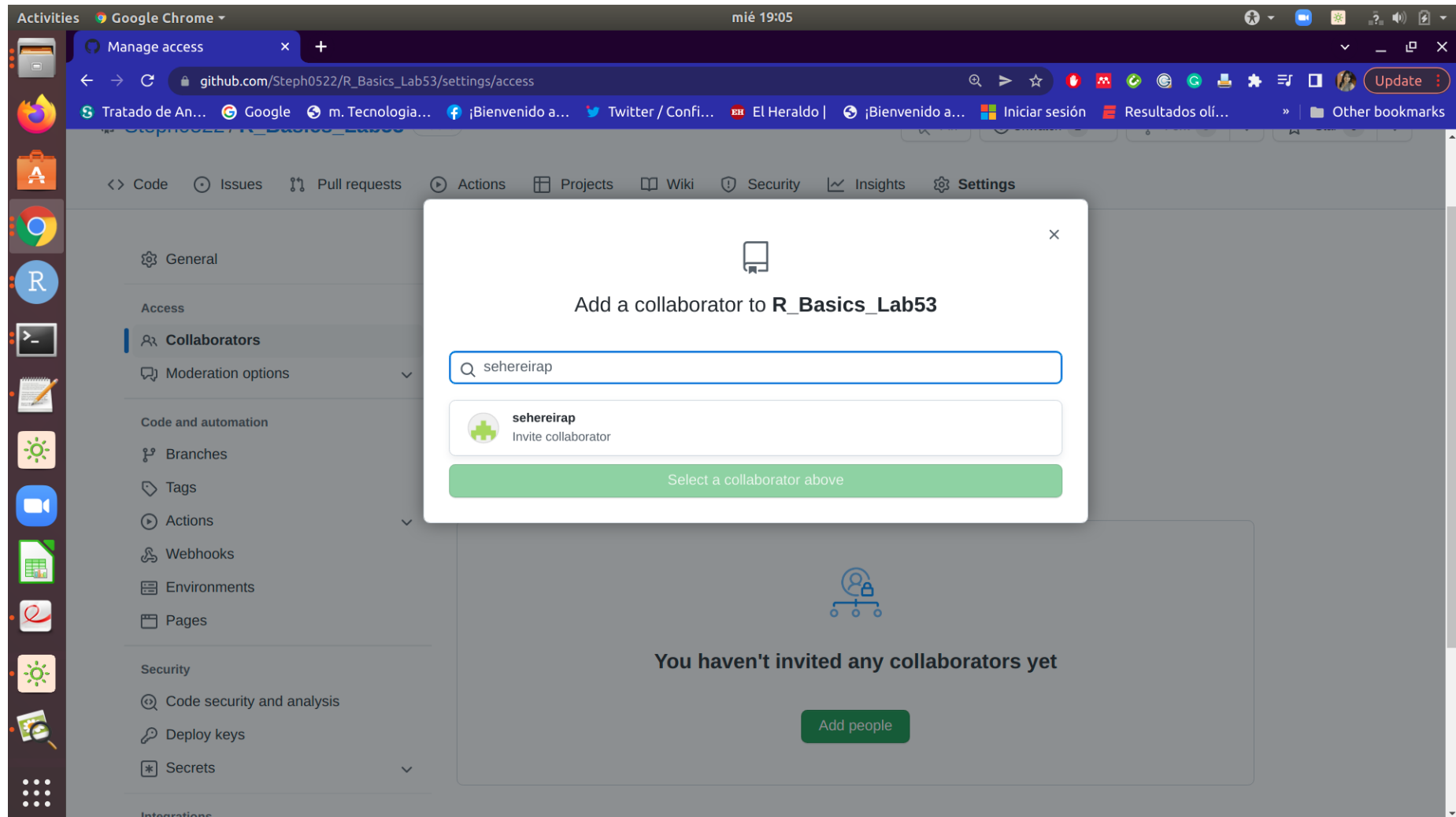
Proyectos colaborativos

- GitHub, Git y Rstudio nos permiten colaborar en nuestros trabajos y/o repositorios.
- Para esto, dentro del repositorio que queremos colaborar le damos *'Settings'* y luego *'Collaborators'* y ponemos el nombre de usuario con quién queremos colaborar y luego lo agregamos.
- El colaborador recibirá un correo de invitación, aceptamos y procedemos a clonar el repositorio para poder introducir cambios y/o actualizaciones.

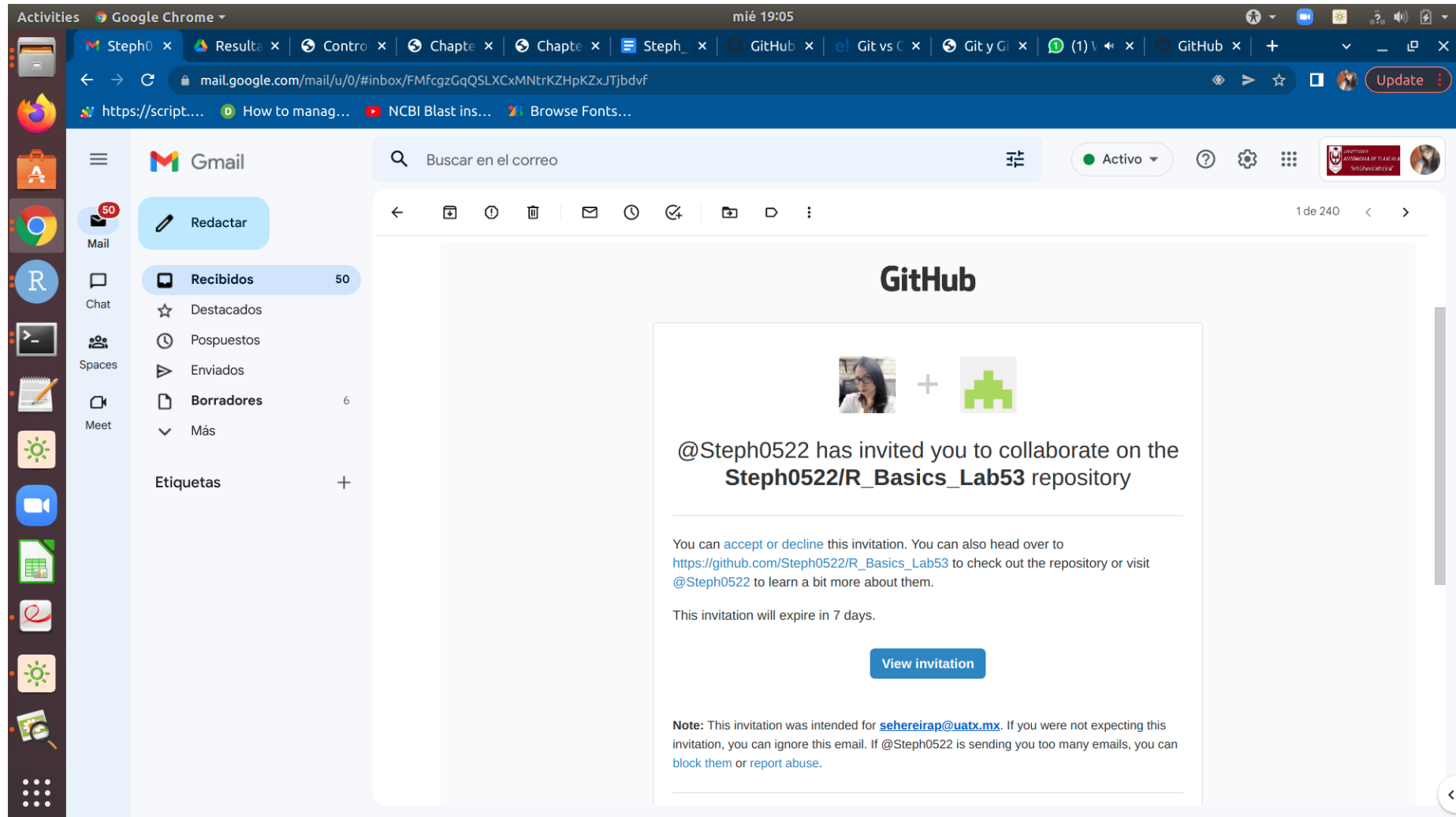
Vamos a configuración



Vamos a colaboradores



Invitación a colaborar



Otros conceptos...

fork: es copiar un repositorio de la cuenta de un usuario cuando no tiene acceso de escritura. Así que puedes simplemente copiarlo y modificarlo bajo tu propia cuenta.

merge: ahora el usuario que es el propietario de esos códigos, si encuentra que sus cambios son relevantes, puede fusionar los cambios encontrados en su repositorio con el repositorio original, simplemente haciendo clic en el botón.

Nota: Si el propietario acepta sus cambios, obtiene su crédito en el sitio original y aparece en su perfil de usuario. Por lo tanto, mientras más personas y proyectos haya en GitHub mejor. Esto anima a los jóvenes desarrolladores a crecer.

¡ Gracias !

Las diapositivas se construyeron con el paquete de R **xaringan**.

Referencias

- <https://happygitwithr.com/index.html>
- <https://github.com/bryan2811/Curso-de-Git-y-Github-Platzi>
- <https://www.edureka.co/blog/git-vs-github/>
- <https://comunidadbioinfo.github.io/cdsb2022/control-de-versiones-con-github-y-rstudio.html>