3- Para el siguiente ejercicio, van a tener que trabajar en equipo, con sus compañeros de mesa.

1. *El facilitador de cada equipo debe crear un repositorio público con el nombre practica\_github seleccionando la opción Initialize this repository with a README.*

Vamos a git hub > nuevo repositorio > le ponemos nombre > acceso público >> archivo Read Me

1. *Una vez creado el repositorio el facilitador debe invitar a los integrantes de su mesa*

*al mismo. Clickear en el botón Invite a collaborator y buscar a los miembros de su*

*mesa por username o email.*

Vamos al repositorio > Settings > Manage Access > Add people > Buscarlo por nombre y poner add

1. *Cada miembro debe aceptar la invitación al repositorio.*

*Checkear la invitación el email y clickear en View Invitation.*

A partir de acá, tienen que ir, los colaboradores, a su mail y aceptar la invitación

1. *Clonar el repositorio.*

*Cada miembro del equipo debe clonar el repositorio con el archivo ReadMe.*

*Luego de aceptar la invitación github te redirige al repositorio.*

Para clonar el repositorio >> Primero copiarás la URL del repositorio remoto que deseas clonar (ver el icono "Copy to clipboard” en la siguiente imagen). Luego abrirás una ventana de terminal, para situarte sobre la carpeta de tu proyecto que quieras clonar. Yo te recomendaría crear ya directamente una carpeta con el nombre del proyecto que estás clonando, o cualquier otro nombre que te parezca mejor para este repositorio. Te sitúas dentro de esa carpeta y desde ella lanzamos el comando para hacer el clon, que sería algo como esto: git clone https://github.com/EggEducacion/MiPrimerRepositorio.git . 21 El último punto, después de la url copiada desde git, le indica que el clon lo vas a colocar en la carpeta donde estás situado, en tu ventana de terminal. La salida de ese comando sería más o menos como tienes en la siguiente imagen: De esta manera nosotros ya tenemos el repositorio remoto para trabajar local y podremos hacer los cambios que queramos y subir los cambios con los comandos que explicamos previamente.

LO DEBEN CLONAR TODOS INCLUIDO EL QUE CREA EL REPOSITORIO

1. *Cada miembro de la mesa, incluido el facilitador, debe crear su propia rama para*

*trabajar sobre el archivo ReadMe*

**Ver en que rama estamos y las ramas existentes**🡪 git branch

Mostrar todas las ramas con los commits🡪 git show-branch

**Crear rama** 🡪 git Branch <nombre rama nueva>

**Pasar de una rama a otra** 🡪 git checkout <nombre rama a la que quiero pasar (ir)>

**Crear rama e ir a la rama que creamos** 🡪 git checkout -b <nombre rama nueva>

**Subir la nueva rama a GitHub** 🡪 git push origin <nombre\_de\_tu\_branch>

\*\*\* Si **no quieres poner siempre origin y el nombre de tu rama** en tus push, tienes que sumarle al push anterior, -u antes de la palabra origin. Esto hará que puedas poner git push solamente y vaya siempre a esa rama. Es importante asegurarse que lo hagamos en una rama nuestra y no en master, ya que podríamos mandar cambios a la rama master pensando que iban a la nuestra. Esto sería asi 🡪 git push -u origin nombre\_de\_tu\_branch 🡪 Una vez esto hecho esto podríamos pararnos en nuestra rama y simplemente escribir 🡪 git push

**Fusionar Ramas (merge)🡪** Nos situamos en una rama, en este caso la "master", y decimos con qué otra rama se debe fusionar el código. El siguiente comando, lanzado desde la rama "master", permite fusionarla con la rama "ramaGit". 🡪 git merge ramaGit

Un merge necesita un mensaje, igual que ocurre con los commit, por lo que al realizar ese comando se abrirá "Vim" (o cualquier otro editor de consola que tengas configurado) para que introduzcas los comentarios que juzgues oportuno. Salir de Vim lo consigues pulsando la tecla ESC y luego escribiendo **:q** y pulsando enter para aceptar ese comando.

Esta operativa de indicar el mensaje se puede resumir con el comando🡪 git merge ramaGit -m “Esto es un merge con mensaje” 🡪 Luego podremos comprobar que nuestra rama master tiene todo el código nuevo de la ramaGit y podremos hacer nuevos commits en master para seguir el desarrollo de nuestro proyecto ya con la rama principal, si es nuestro deseo.

**Borrar rama** 🡪 git branch -D <branch-name>

1. *Ahora cada miembro de la mesa, debe incluir su nombre en el archivo ReadMe de*

*manera local y subirlo a su rama.*

Modificamos el readme en la compu

Depsues en la consola de git 🡪 git add . 🡪 git commit -m “mensaje” 🡪 git push origin <rama actual>

1. *Cuando todos los miembros de la mesa han agregado su nombre al archivo ReadMe,*

*de uno en uno, ir uniendo en la rama master todos vuestros cambios. Al final les*

*debería quedar un archivo ReadMe con todos sus nombres.*

*4 - Ahora van a continuar trabajando como mesa. Vuestra tarea ahora es que cada miembro*

*de la mesa, incluido el facilitador, debe crear su branch y crear una de las siguientes*

*clases: Gato, Perro, Caballo, Conejo, Pájaro y Pato. Cada uno le va a poner los atributos*

*que desee.*

*El facilitador va a tener que crear el repositorio y subir un proyecto de Java vacio para*

*que los miembros de la mesa puedan clonar y crear su clase. Una vez que cada miembro*

*haya creado su clase en su respectiva rama, deberán unir todas las clases en la rama*

*master, para que quede el proyecto final*