battlebots

EI2001 Taller de Proyecto

María José Alfaro, Fernanda Sanchirico, Salvador Alveal
(ft. Felipe Higuera, Gustavo Holmberg, Cristóbal Mesías)
Matías Mattamala

Otoño 2019

agenda

1. qué se viene
2. tutorial de git

	L	M	M	J	V	S	D	
semana 11		28	29	30	31	1	2	
semana 12	3	4	**5**	6	7	8	9	junio
semana 13	10	11	**12**	13	14	15	16	
semana 14	17	18	**19**	20	21	22	23	
semana 15	24	25	**26**	27	28	29	30	
	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	julio
	15	16	17	18	19	20	21	
	22	23	24	25	26	27	28	
	29	30	31	1	2	3	4	
	5	6	7	8	9	10	11	agosto

	L	M	M	J	V	S	D	
semana 11		28	29	**30**	31	1	2	
semana 12	3	4	5	**6**	7	8	9	junio
semana 13	10	11	12	**13**	14	15	16	
semana 14	17	18	19	**20**	21	22	23	
semana 15	24	25	26	**27**	28	29	30	
	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	julio
	15	16	17	18	19	20	21	
	22	23	24	25	26	27	28	
	29	30	31	1	2	3	4	
	5	6	7	8	9	10	11	agosto

	L	M	M	J	V	S	D	
semana 11		28	29	**30**	31	1	2	
semana 12	3	4	5	**6**	7	8	9	junio
semana 13	10	11	12	**13**	14	15	16	
semana 14	17	18	19	**20**	21	22	23	
semana 15	24	25	26	**27**	28	29	30	
	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	julio
	15	16	17	18	19	20	21	
	22	23	24	25	26	27	28	
	29	30	31	1	2	3	4	
	5	6	7	8	9	10	11	agosto

	L	M	M	J	V	S	D	
semana 11		28	29	**30**	31	1	2	
semana 12	3	4	5	**6**	7	8	9	junio
semana 13	10	11	12	**13**	14	15	16	
semana 14	17	18	19	**20**	21	22	23	
semana 15	24	25	26	**27**	28	29	30	
	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	julio
	15	16	17	18	19	20	21	
	22	23	24	25	26	27	28	
	29	30	31	1	2	3	4	
	5	6	7	8	9	10	11	agosto

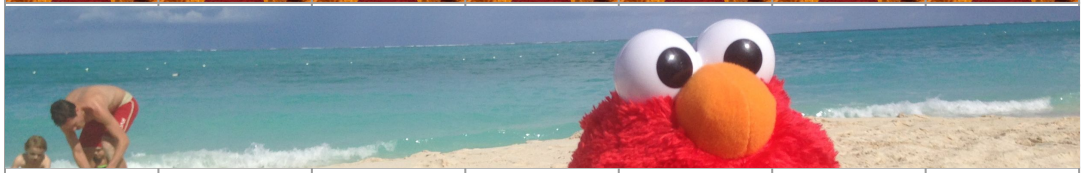
	L	M	M	J	V	S	D	
semana 11		28	29	**30**	31	1	2	
semana 12	3	4	5	**6**	7	8	9	junio
semana 13	10	11	12	**13**	14	15	16	
semana 14	17	18	19	**20**	21	22	23	
semana 15	24	25	26	**27**	28	29	30	
	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	julio
	15	16	17	18	19	20	21	
	22	23	24	25	26	27	28	
	29	30	31	1	2	3	4	
	5	6	7	8	9	10	11	agosto

	L	M	M	J	V	S	D
semana 11		28	29	**30**	31	1	2
semana 12	3	4	5	**6**	7	8	9
semana 13	10	11	12	**13**	14	15	16
semana 14	17	18	19	**20**	21	22	23
semana 15	24	25	26	**27**	28	29	30
	1	2	3	4	5	6	7
	8	9	10	11	12	13	14
	29	30	31	1	2	3	4
	5	6	7	8	9	10	11

junio

julio

agosto

	L	M	M	J	V	S	D	
semana 11		28	29	**30**	31	1	2	
semana 12 tutorial git	3	4	5	**6**	7	8	9	junio
semana 13 tutorial esquemas	10	11	12	**13**	14	15	16	
semana 14	17	18	19	**20**	21	22	23	
semana 15 presentaciones	24	25	26	**27**	28	29	30	
	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	julio
								
	29	30	31	1	2	3	4	
	5	6	7	8	9	10	11	agosto

qué se viene

1. evaluar mejoras en sus battlebots
    a. rediseñar armas
    b. incluir sensores
    c. cambiar sistemas de radiocontrol
2. documentar
    a. todo en github
    b. diagramas eléctricos
    c. diagramas funcionales
    d. documentar código

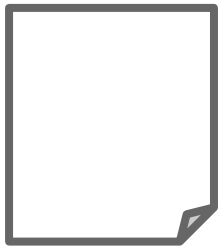
tutorial de git
(al fin)

Tutorial de Git

Control de versiones, Git, GitHub

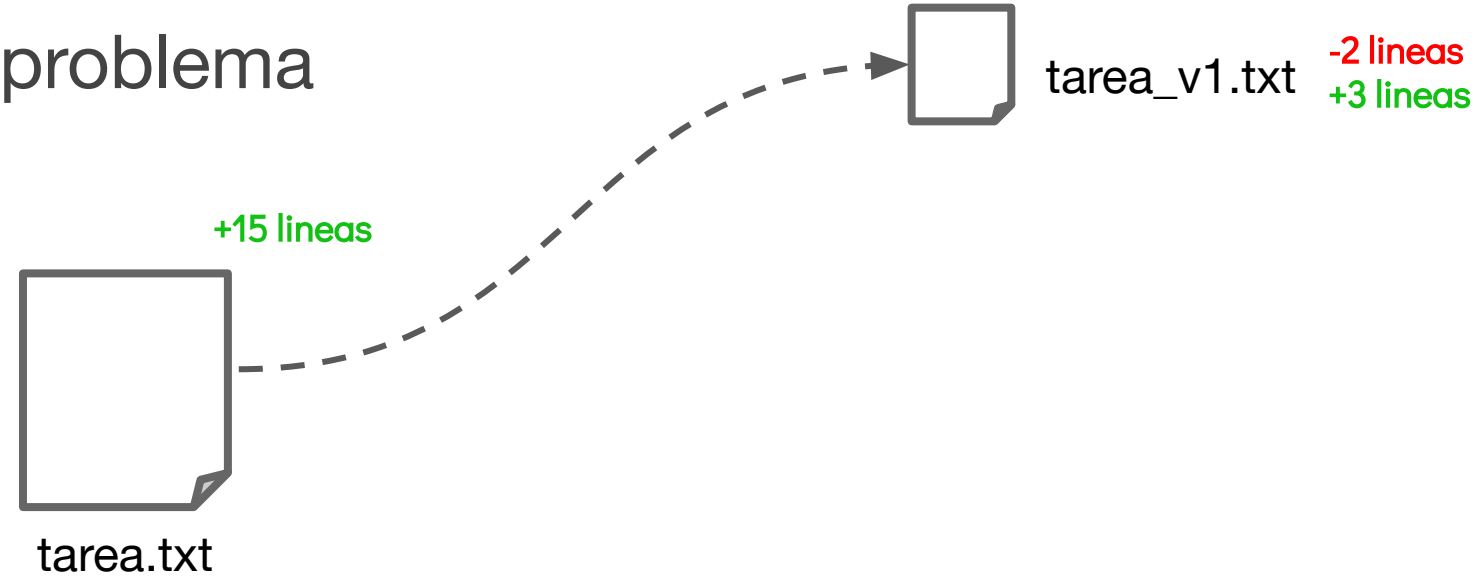
Matías Mattamala
Junio 2019

El problema

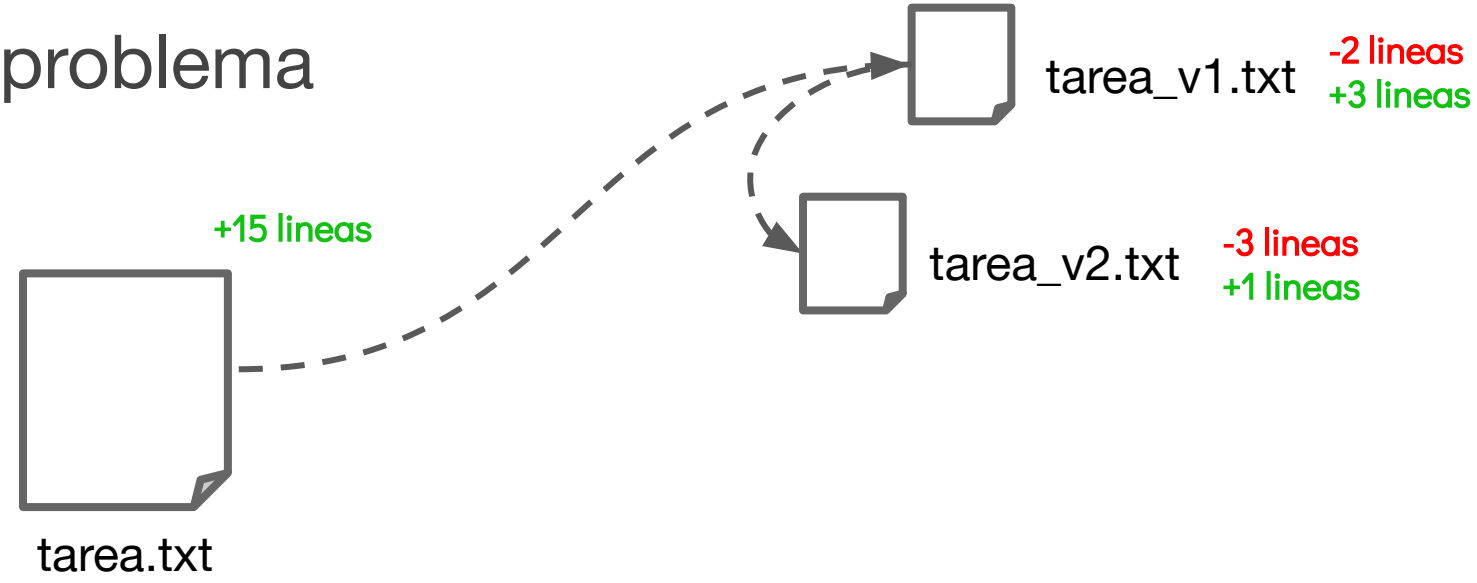


tarea.txt

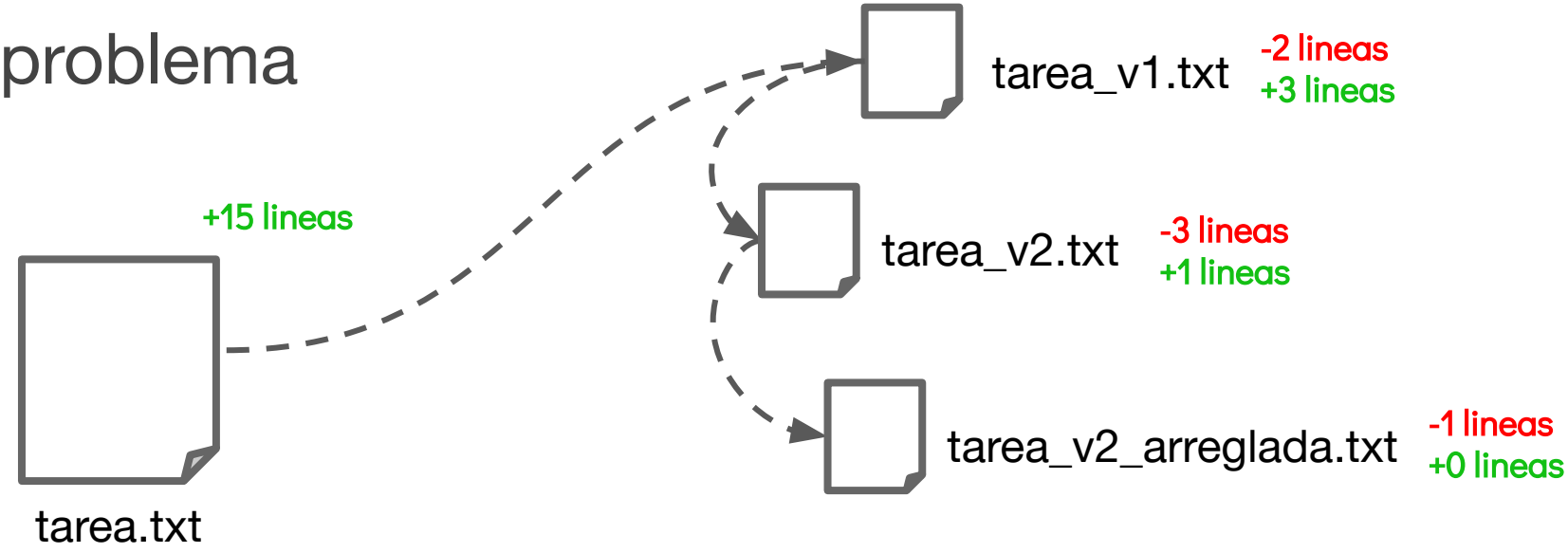
El problema



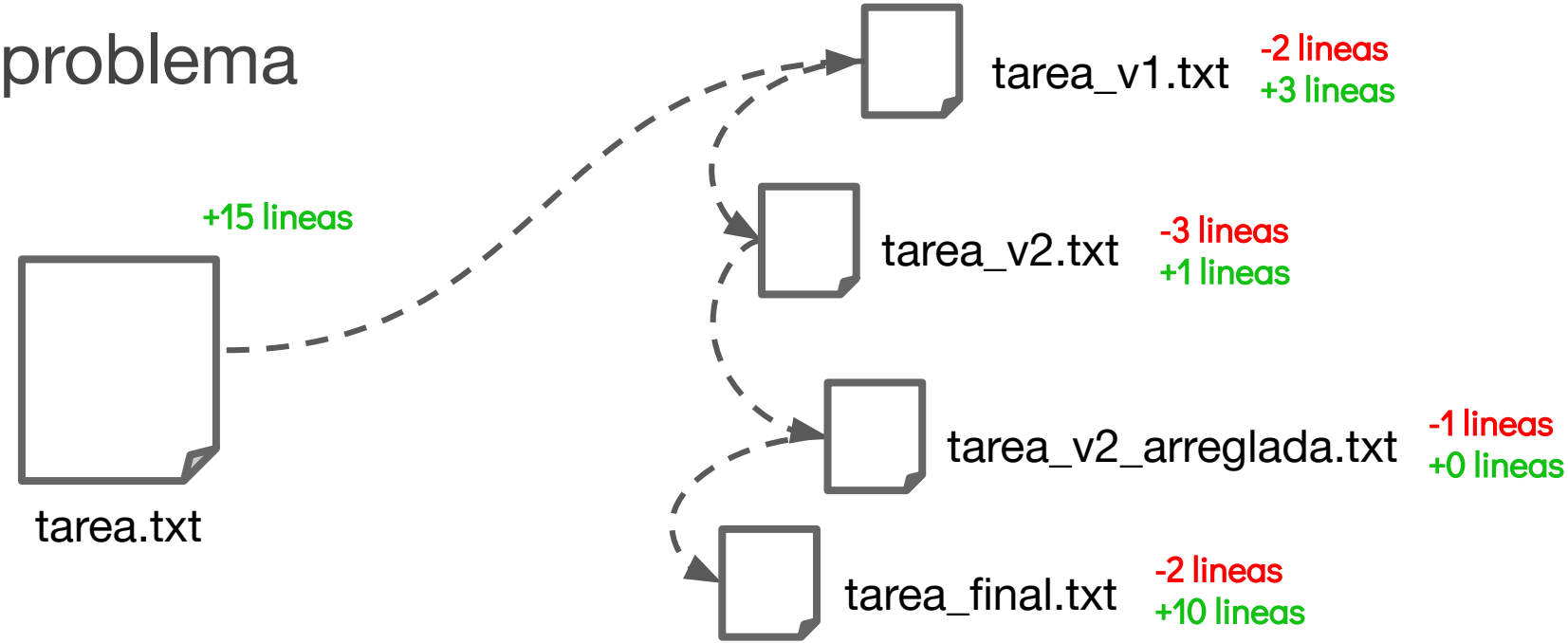
El problema


tarea_v1.txt
-2 lineas
+3 lineas
+15 lineas
tarea_v2.txt
-3 lineas
+1 lineas
tarea.txt

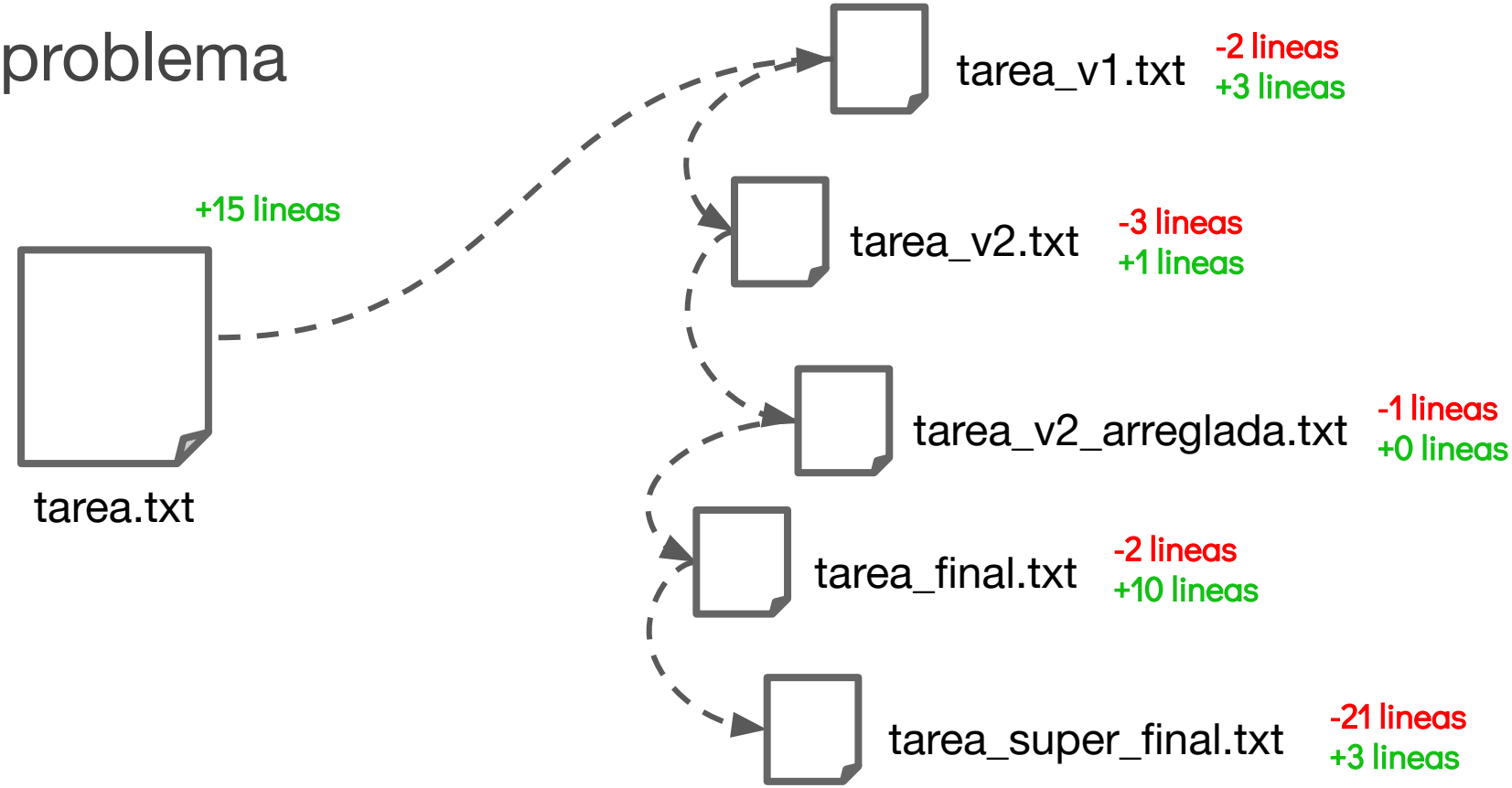
El problema


+15 lineas
tarea.txt
tarea_v1.txt
-2 lineas
+3 lineas
tarea_v2.txt
-3 lineas
+1 lineas
tarea_v2_arreglada.txt
-1 lineas
+0 lineas

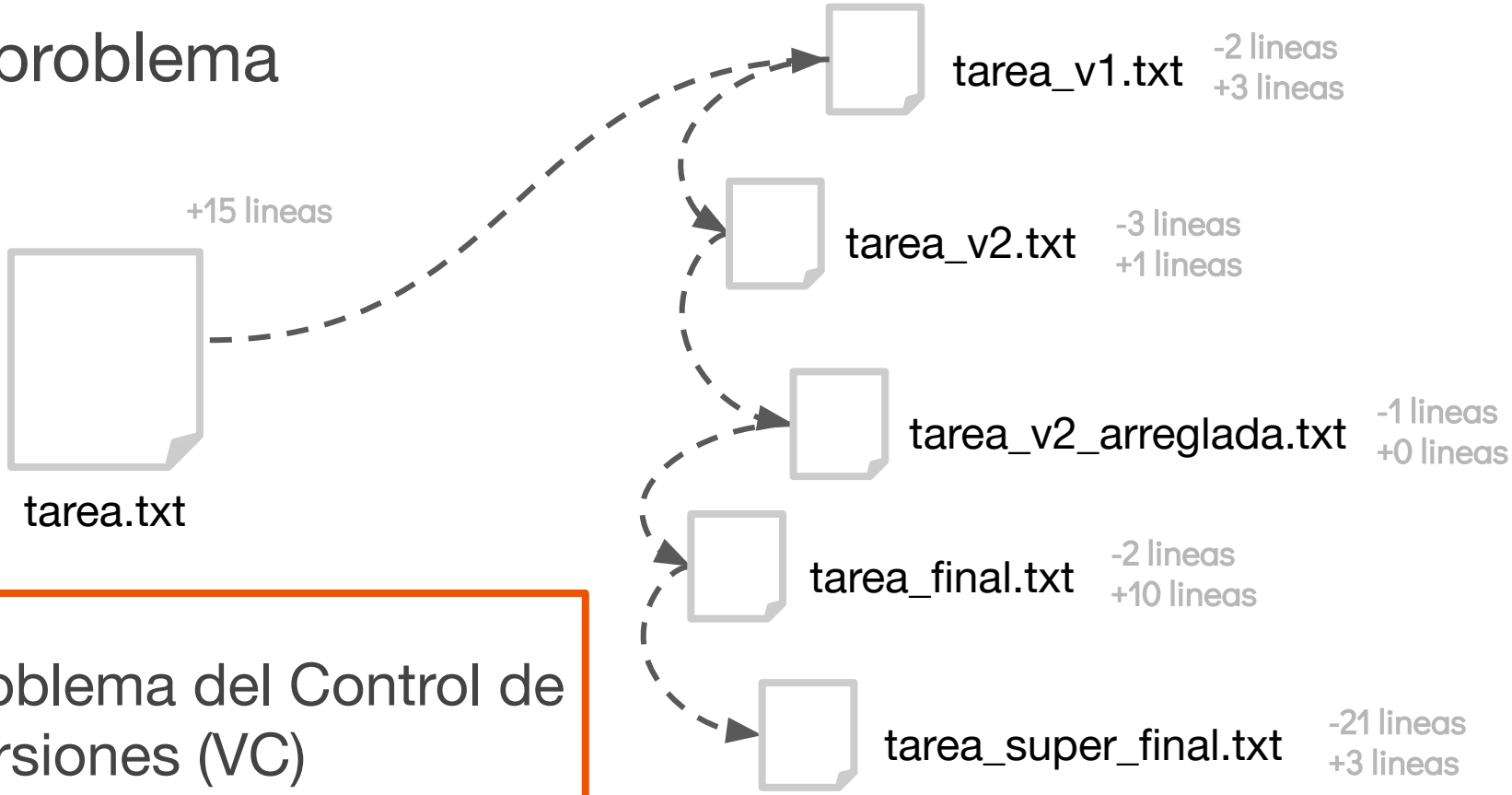
El problema


+15 lineas
tarea.txt
tarea_v1.txt
-2 lineas
+3 lineas
tarea_v2.txt
-3 lineas
+1 lineas
tarea_v2_arreglada.txt
-1 lineas
+0 lineas
tarea_final.txt
-2 lineas
+10 lineas

El problema


+15 lineas
tarea.txt
tarea_v1.txt
-2 lineas
+3 lineas
tarea_v2.txt
-3 lineas
+1 lineas
tarea_v2_arreglada.txt
-1 lineas
+0 lineas
tarea_final.txt
-2 lineas
+10 lineas
tarea_super_final.txt
-21 lineas
+3 lineas

El problema



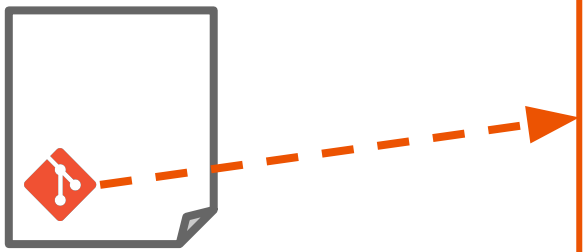
Problema del Control de Versiones (VC)

Git soluciona el problema de Control de Versiones

Sistema de Control de Versiones

*Hay más sistemas (SVN, Mercurial), pero Git es usado por Microsoft, Google, Facebook, Netflix, etc

Qué hace Git

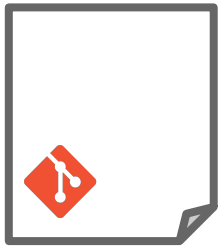


tarea.txt

Historial de versiones

versión	archivo	cambios
V1	tarea_v1.txt	-2 lineas +3 lineas
V2	tarea_v2.txt	-3 lineas +1 lineas
V3	tarea_v2_arreglada.txt	-1 lineas +0 lineas
V4	tarea_final.txt	-2 lineas +10 lineas
V5	tarea_super_final.txt	-21 lineas +3 lineas

Ventajas de usar Git



tarea.txt

1. Administración ordenada de **respaldos**

2. Permite la **colaboración** de varias personas

3. Es **gratis**, así que sólo hay que aprender a usarlo

Aspectos prácticos de Git

Quiero usar Git, ¿cómo lo hago?

1. **Carpeta con código** que queremos respaldar

En jerga de git, es lo que conformará un **repositorio**

Quiero usar Git, ¿cómo lo hago?

1. **Carpeta con código** que queremos respaldar

En jerga de git, es lo que conformará un **repositorio**

2. **Git** (software)

Descargarlo e instalarlo de [git-scm.com](git-scm.com)

Quiero usar Git, ¿cómo lo hago?

1. **Carpeta con código** que queremos respaldar

En jerga de git, es lo que conformará un **repositorio**

2. **Git** (software)

Descargarlo e instalarlo de [git-scm.com](git-scm.com)

3. **Decidir cómo usarlo**

Decidir si se usará Git de forma **local** o **remota**

Uso de Git: Local

El repositorio queda **guardado en nuestros computadores**

Se debe **configurar manualmente** el repositorio

No requiere conexión a internet

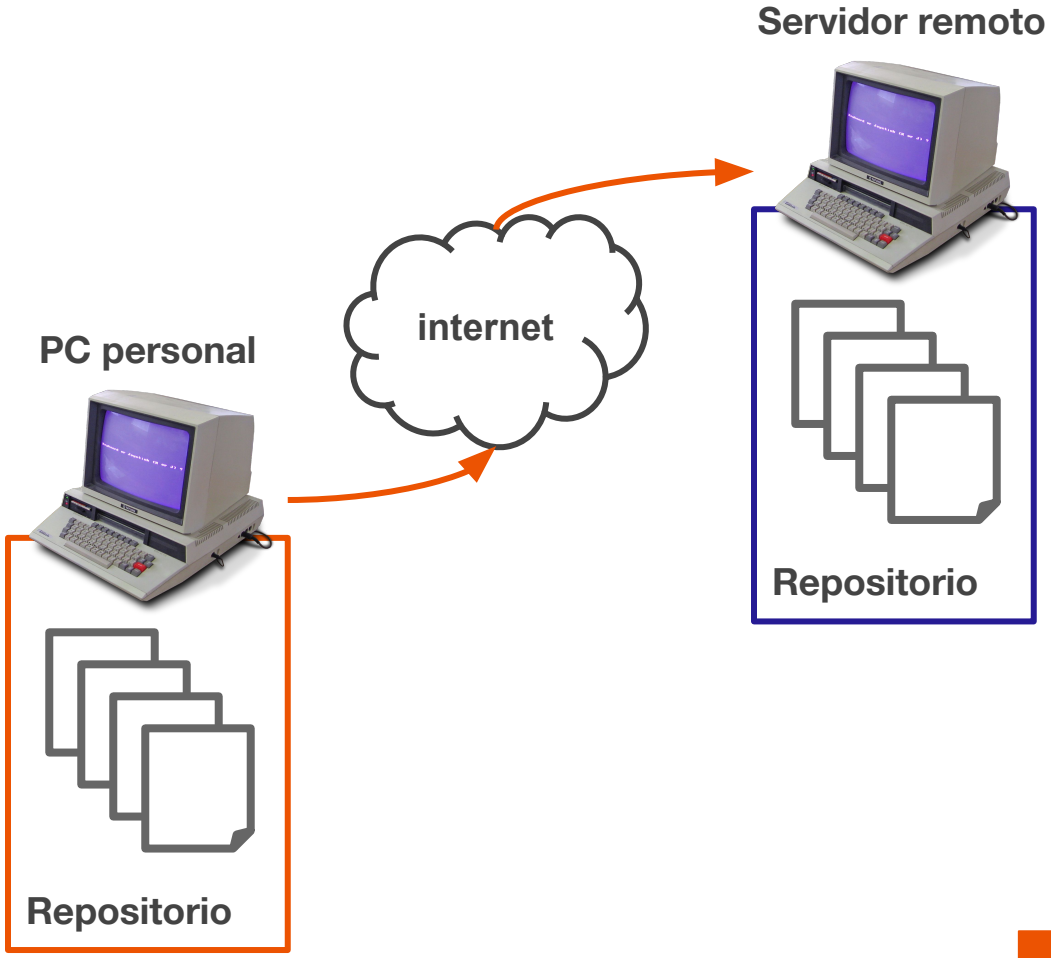
Uso de Git: Remoto

El repositorio queda guardado en **nuestros pcs** y en **servidor remoto**

Podemos **configurar el repositorio remotamente** y **clonarlo** en nuestros pcs

Requiere conexión a internet

Método recomendado



Uso de Git con un servidor remoto

Servidores remotos de Git

Popular en general

A veces requerido en **entrevistas laborales**

Repositorios:
- públicos por defecto
- privados con cuenta educacional

Popular en **proyectos empresariales**

Repositorios:
- públicos y privados

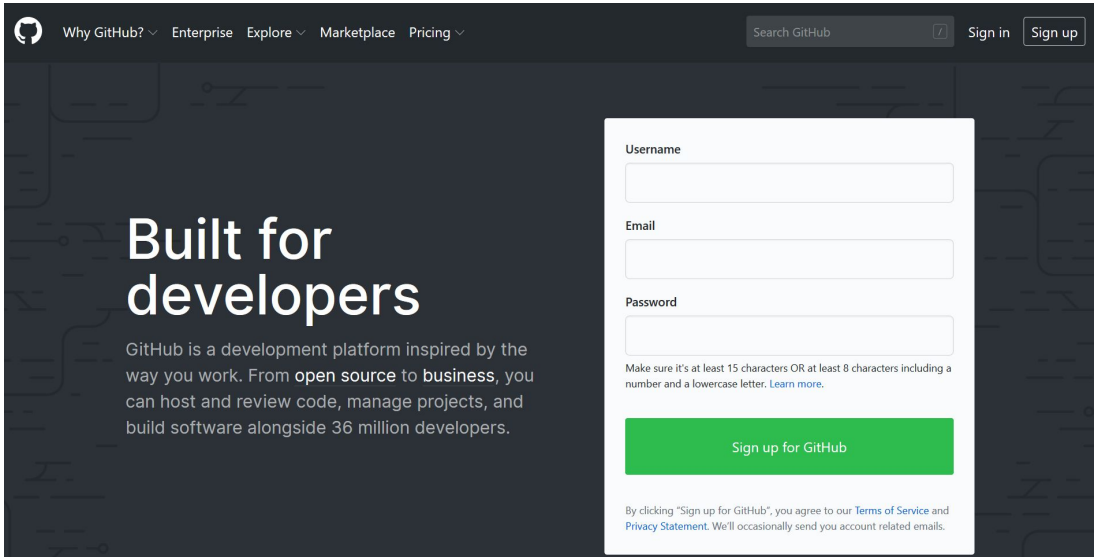
Popular como **alternativa abierta a GitHub**

Repositorios:
- públicos por defecto
- privados (con limitaciones)

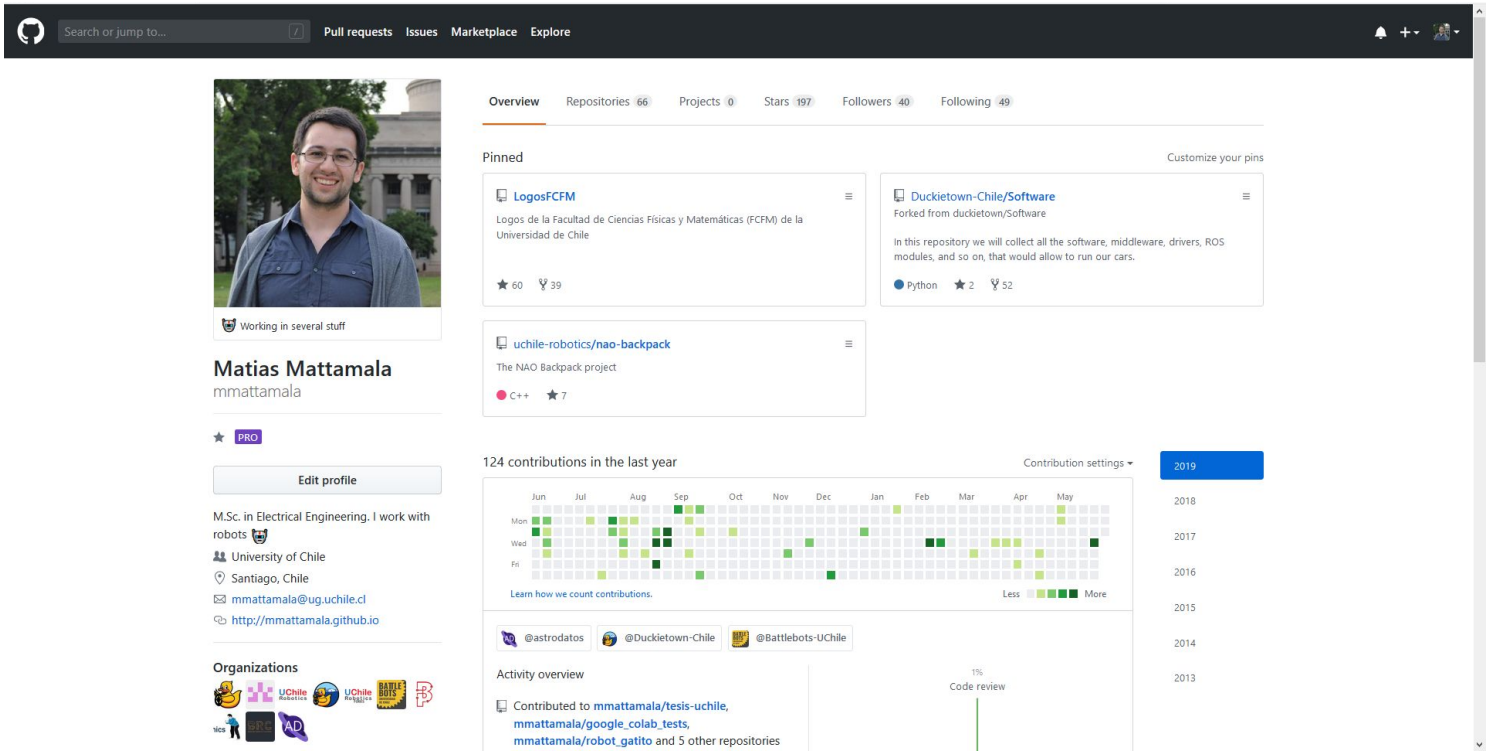
Caso de estudio: GitHub

GitHub requiere **crear un usuario** en su sitio www.github.com

El usuario da acceso a **crear repositorios**, participar en **proyectos** y **organizaciones**, y tener un **perfil público**



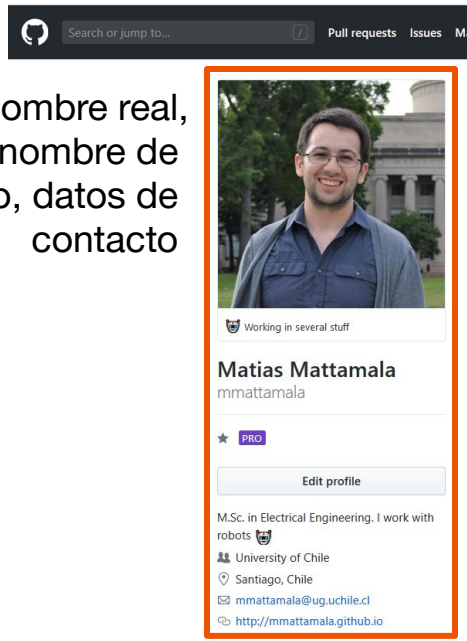
GitHub: Perfiles de usuario



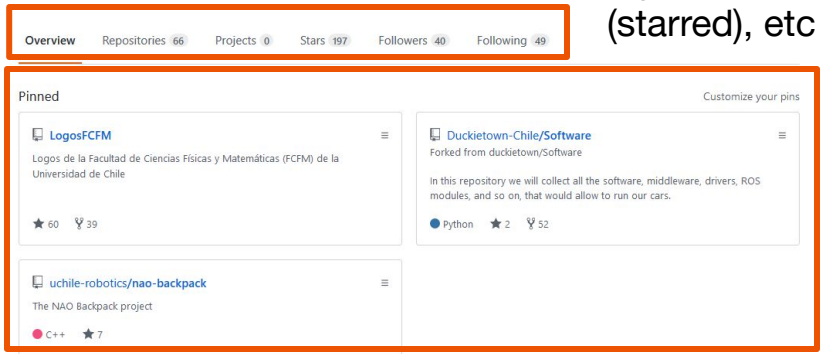
https://github.com/mmattamala

GitHub: Perfiles de usuario

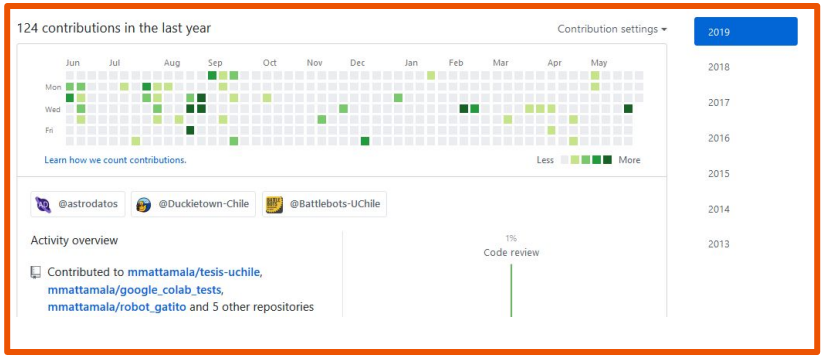
Foto, nombre real, nombre de usuario, datos de contacto



Repositorios, proyectos, repositorios destacados (starred), etc



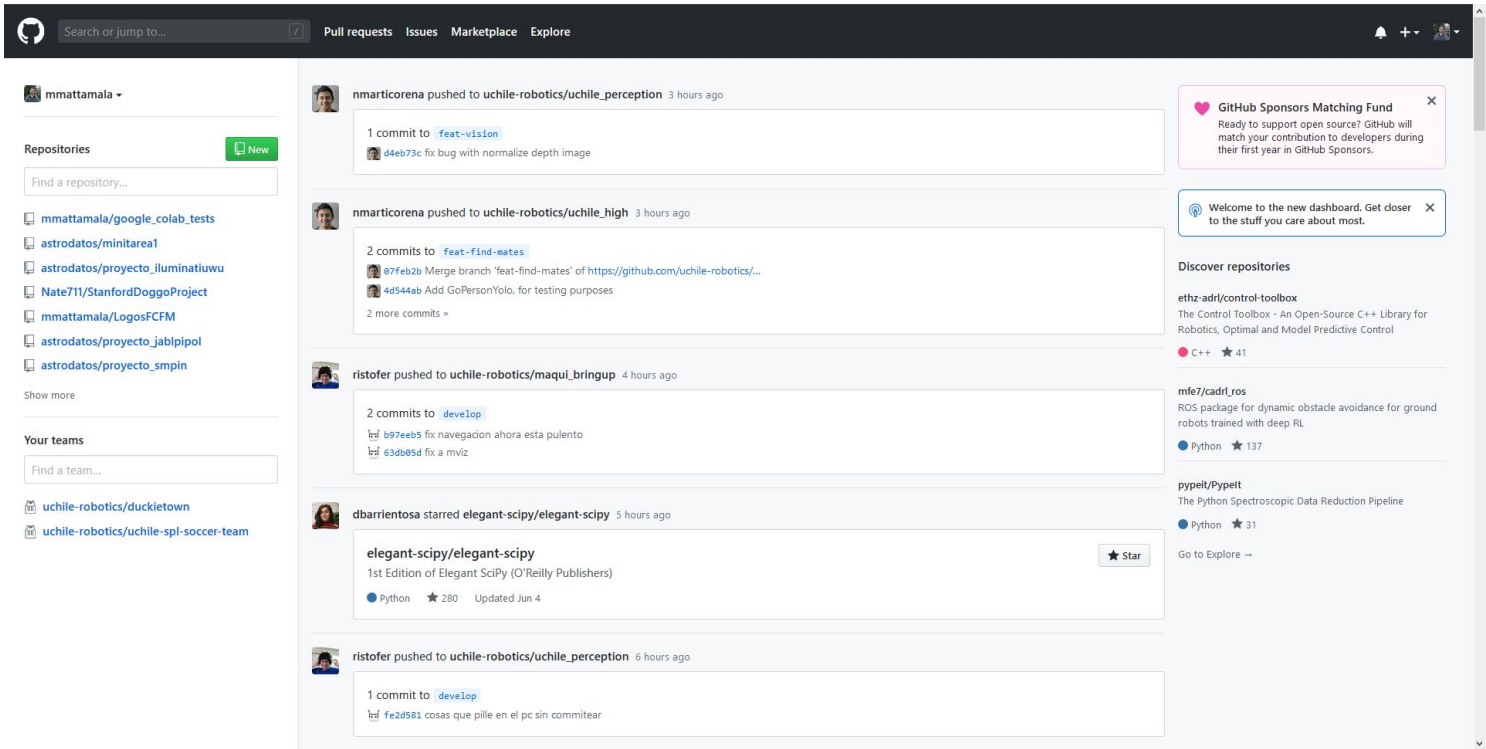
Contribuciones



Organizaciones asociadas

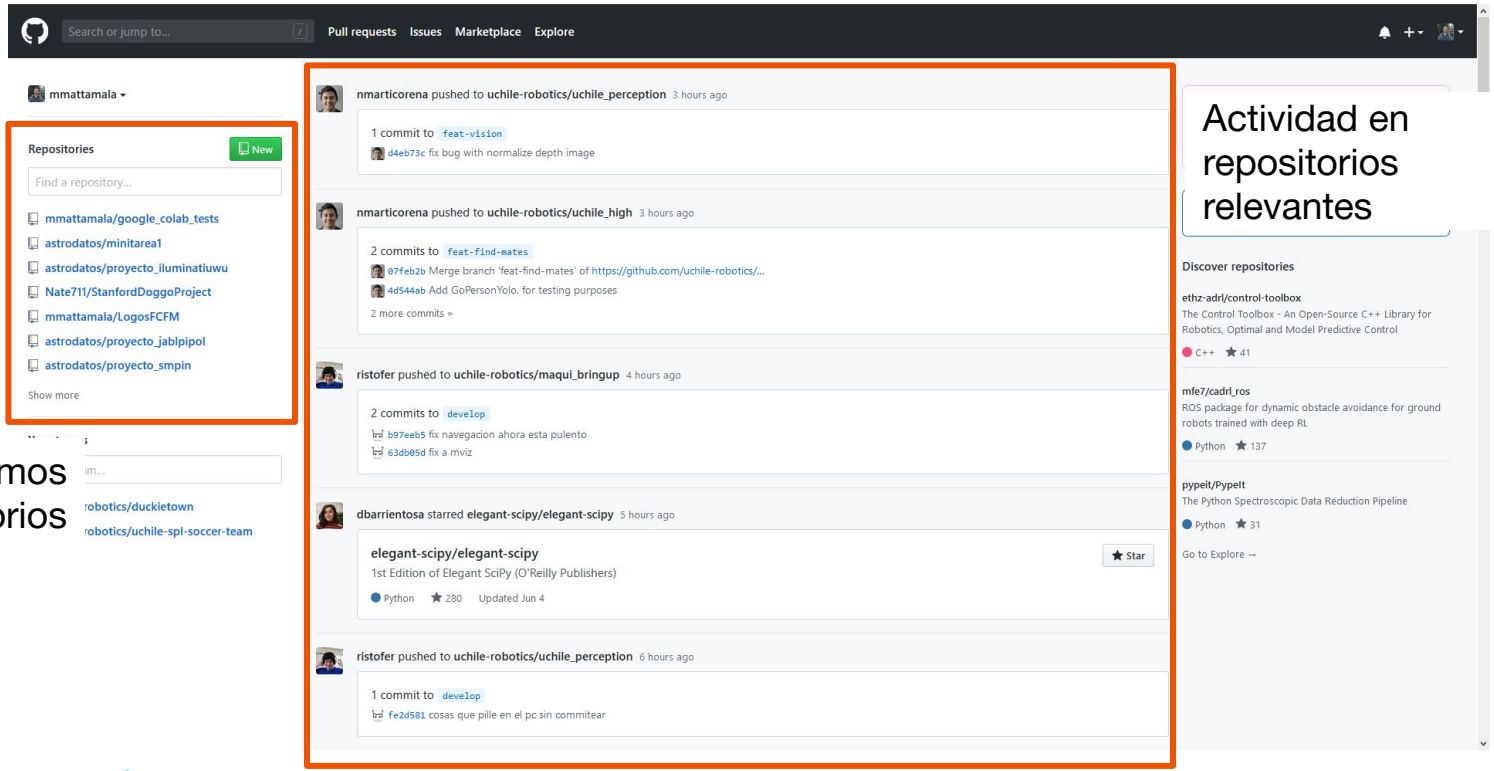
https://github.com/mmattamala

GitHub: Dashboard



https://github.com/

GitHub: Dashboard

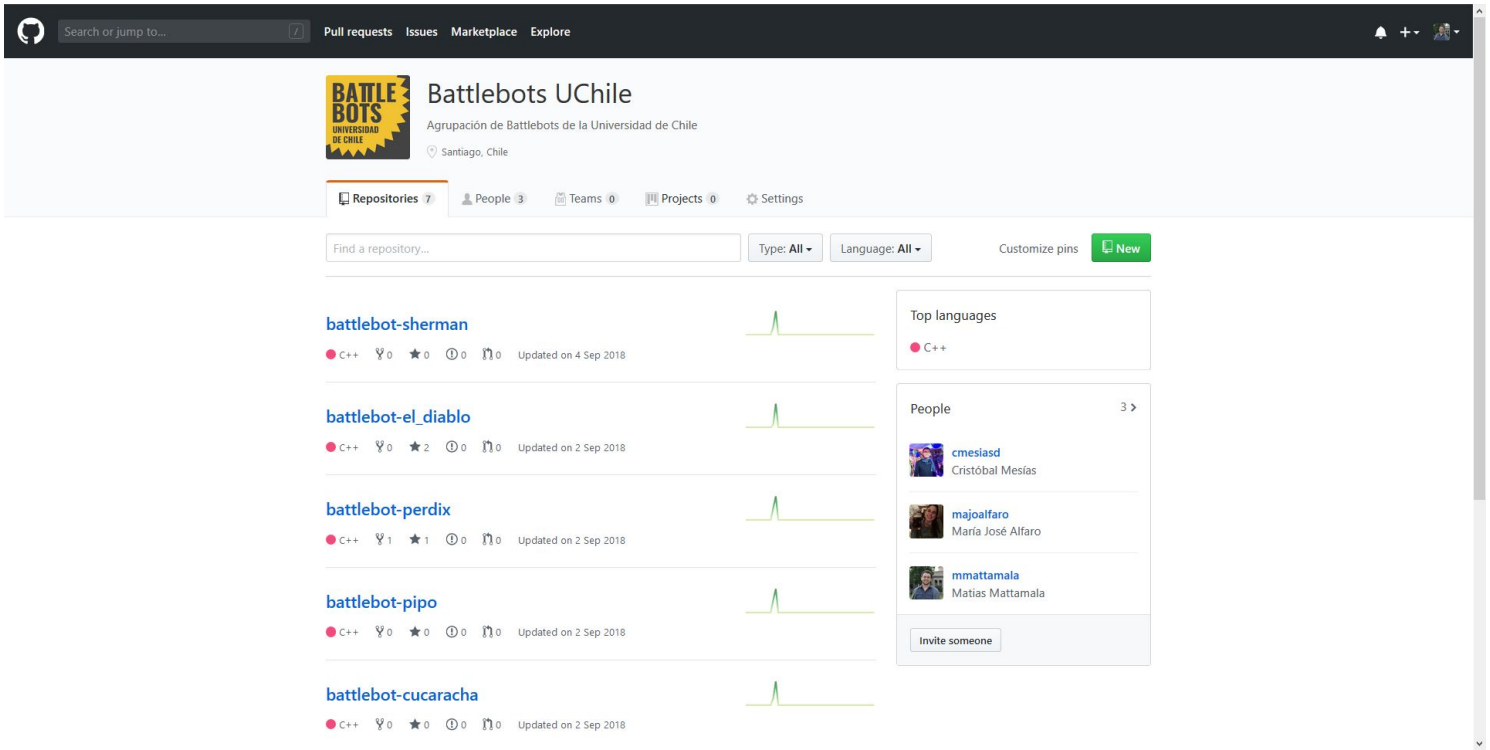


Actividad en repositorios relevantes

Últimos repositorios

https://github.com/

GitHub: Organizaciones



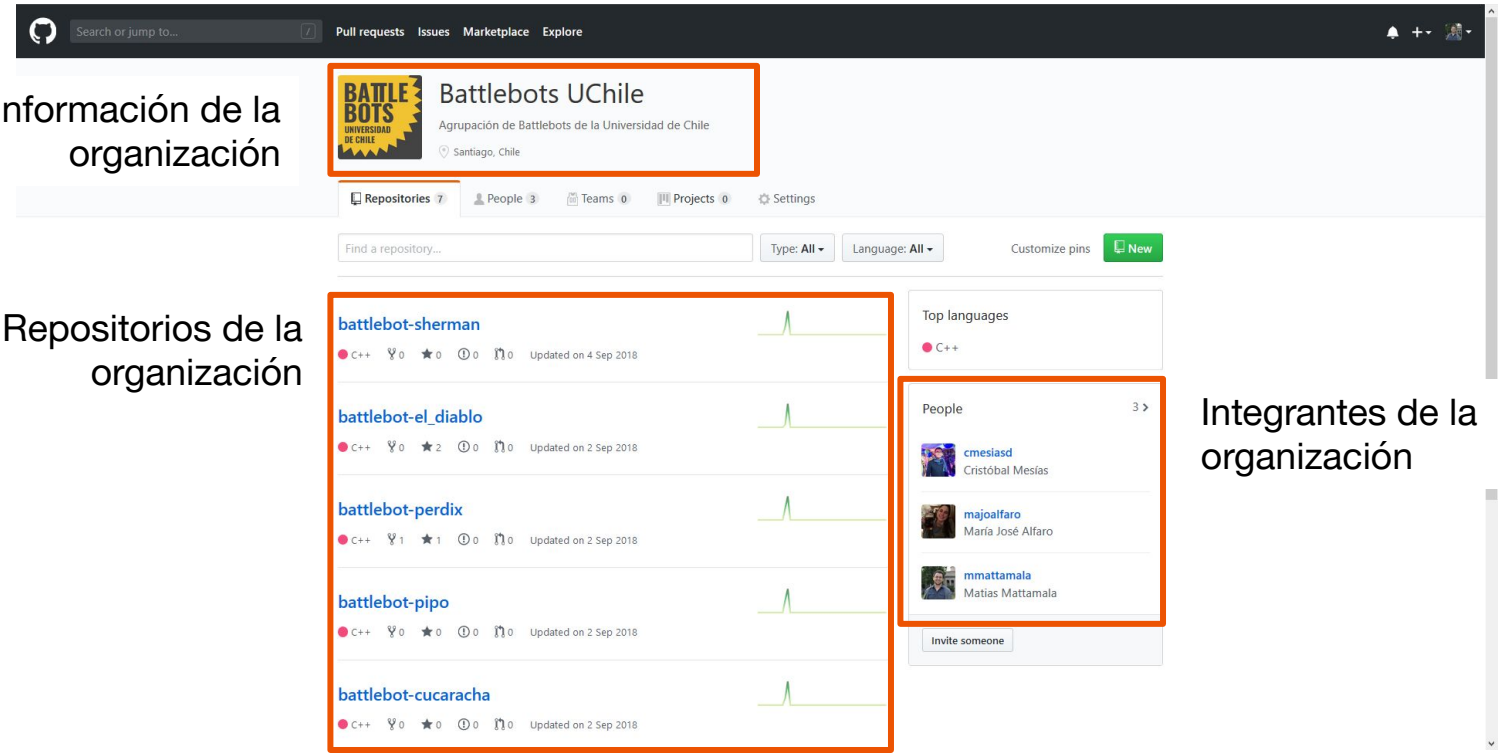
https://github.com/Battlebots-UChile/

GitHub: Organizaciones

Información de la organización

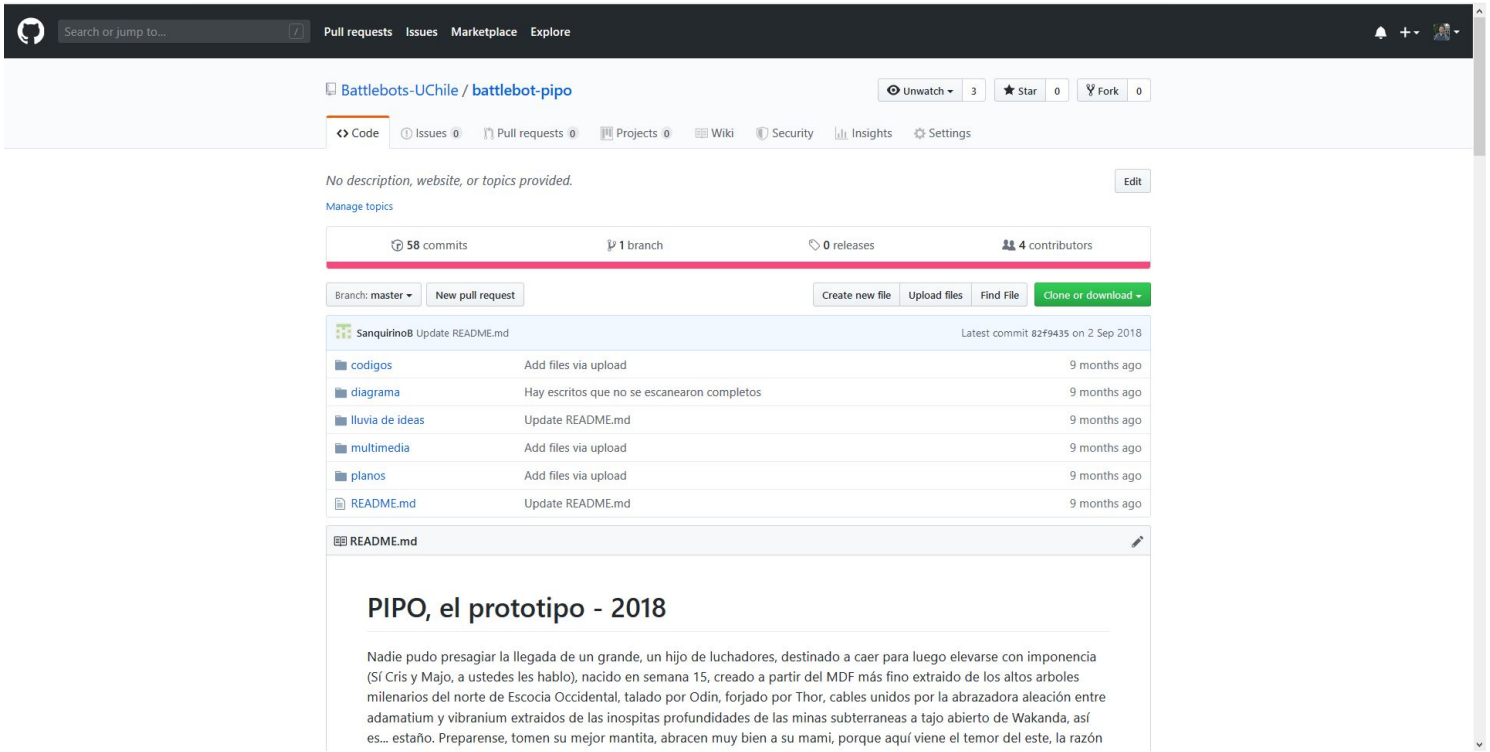
Repositorios de la organización

Integrantes de la organización



https://github.com/Battlebots-UChile/

GitHub: Repositorios



https://github.com/Battlebots-UChile/battlebot-pipo

GitHub: Repositorios

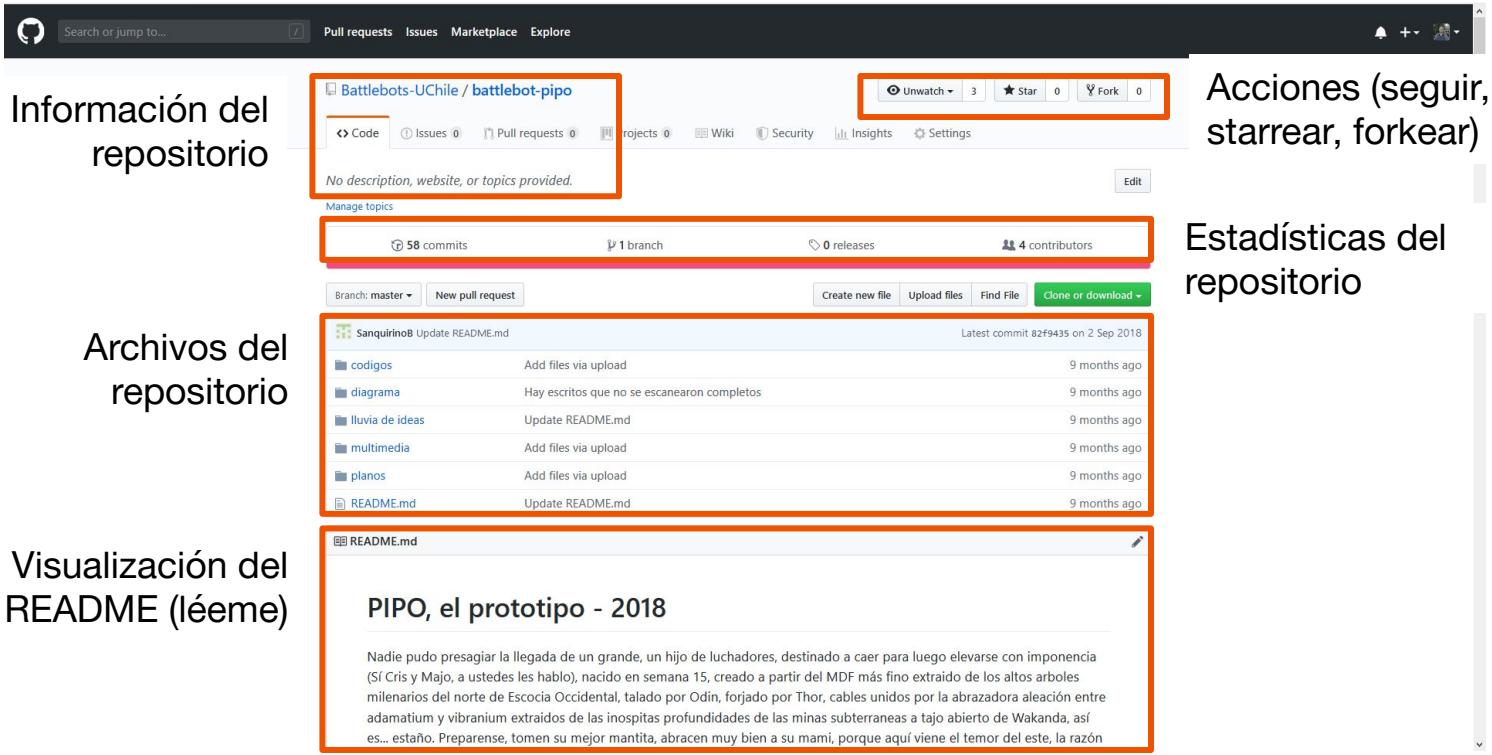
Información del repositorio

Acciones (seguir, starrear, forkear)

Estadísticas del repositorio

Archivos del repositorio

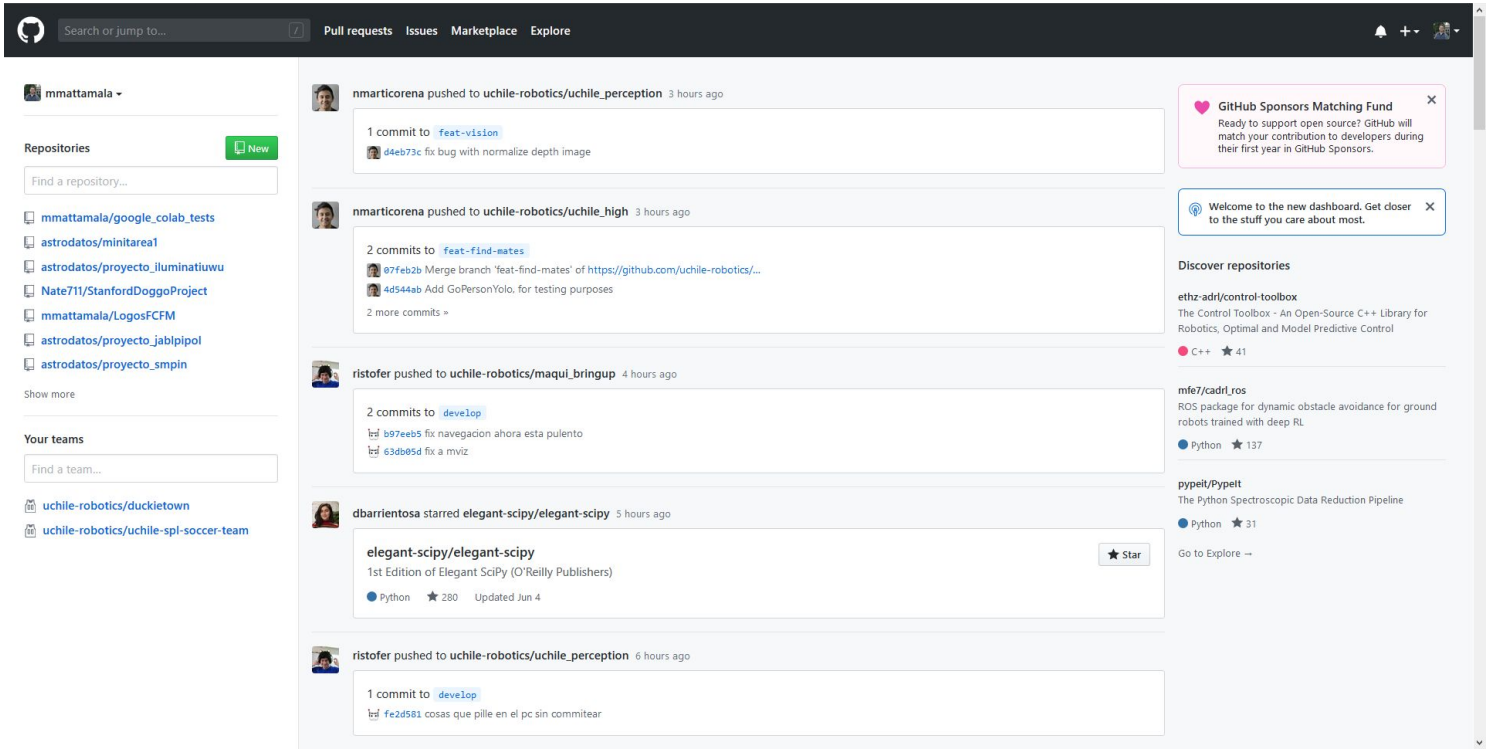
Visualización del README (léeme)



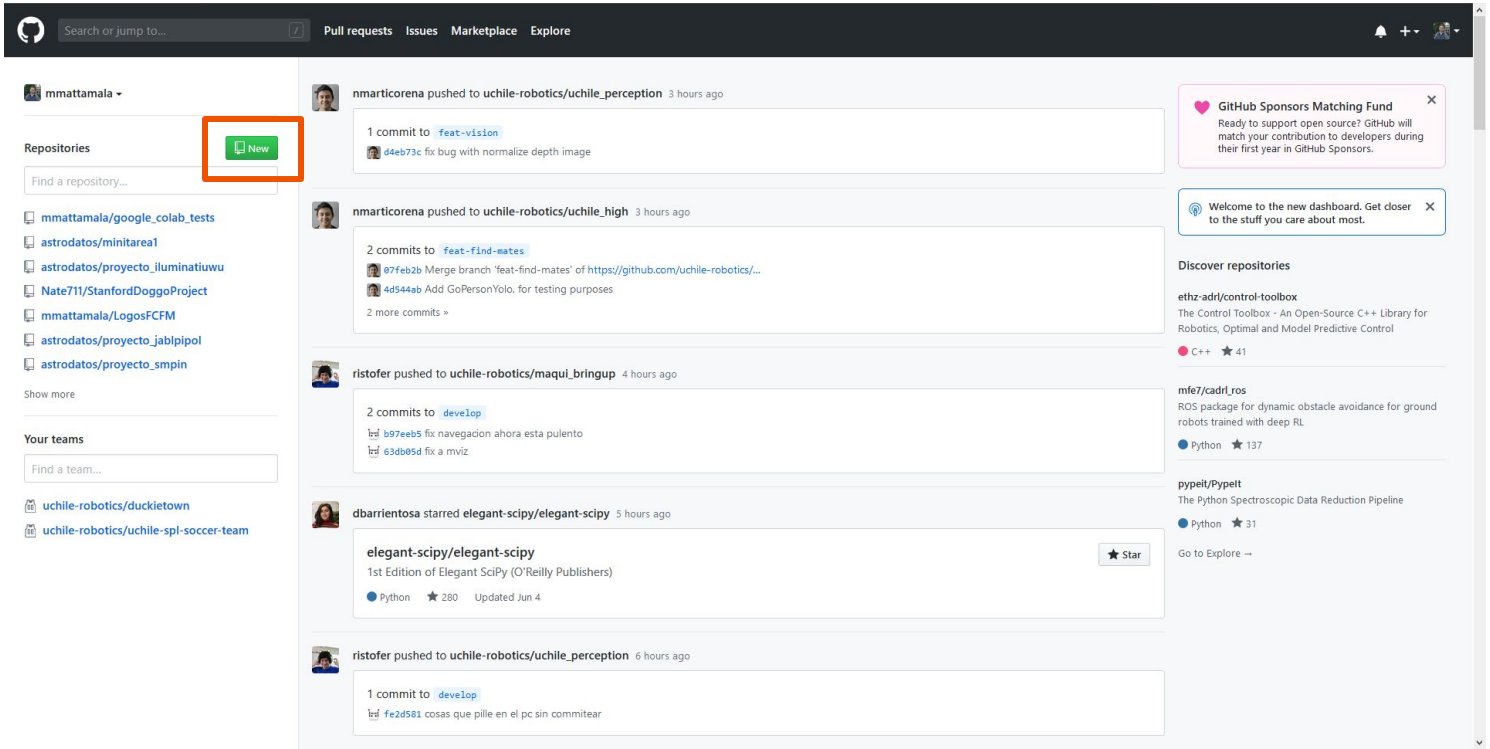
https://github.com/Battlebots-UChile/battlebot-pipo

Creando repositorios en GitHub

Paso 1: Ir al Dashboard

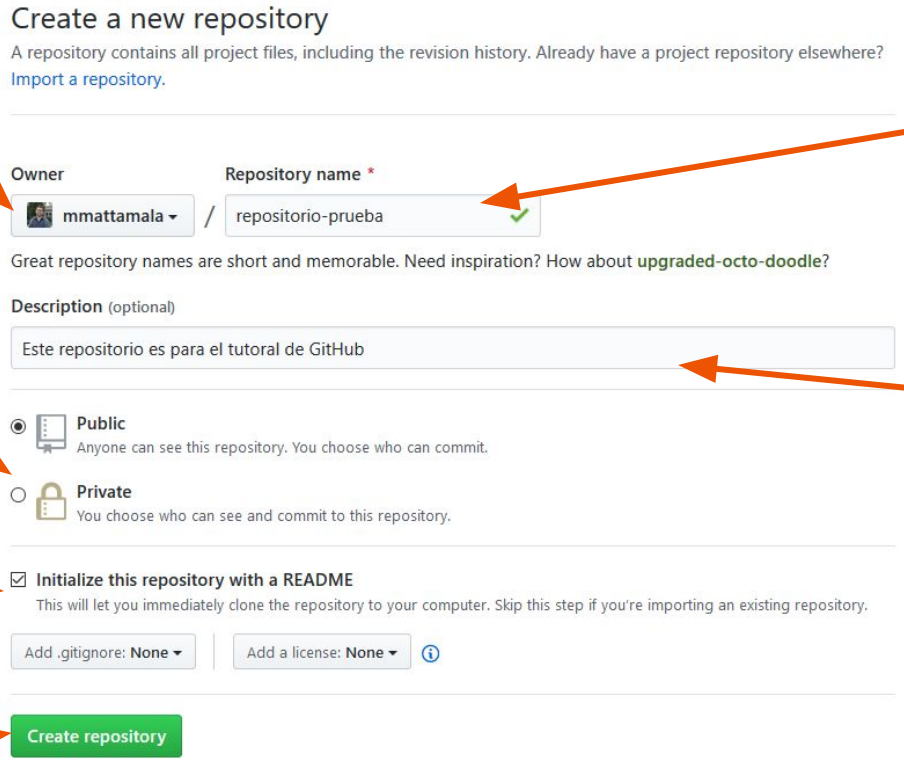


Paso 2: Seleccionar “New”



Paso 3: Configurar opciones básicas

Dónde se va a alojar el repositorio (**usuario u organización**)



Nombre del repositorio (sin espacios)

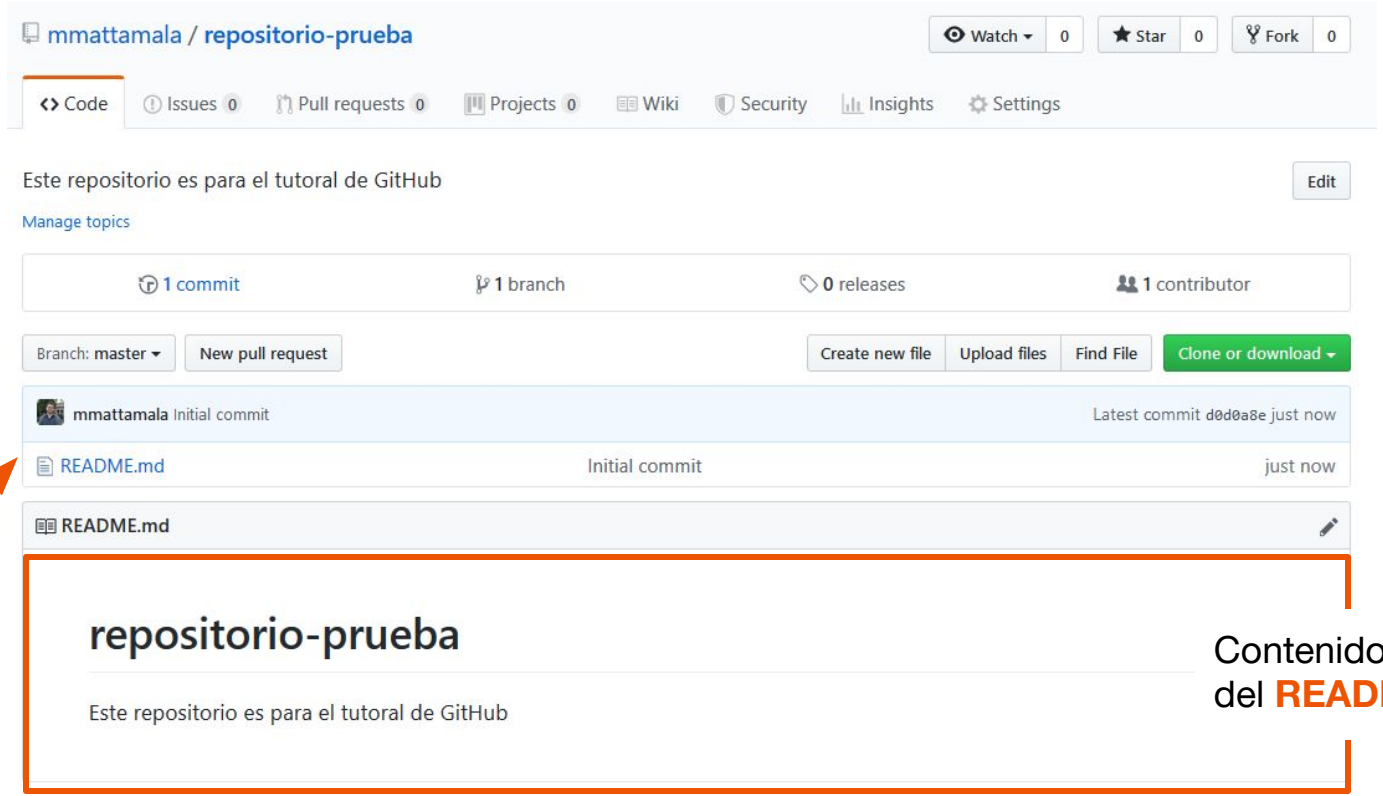
Descripción (recomendada)

Público o privado

Inicializar con **README**

Crear

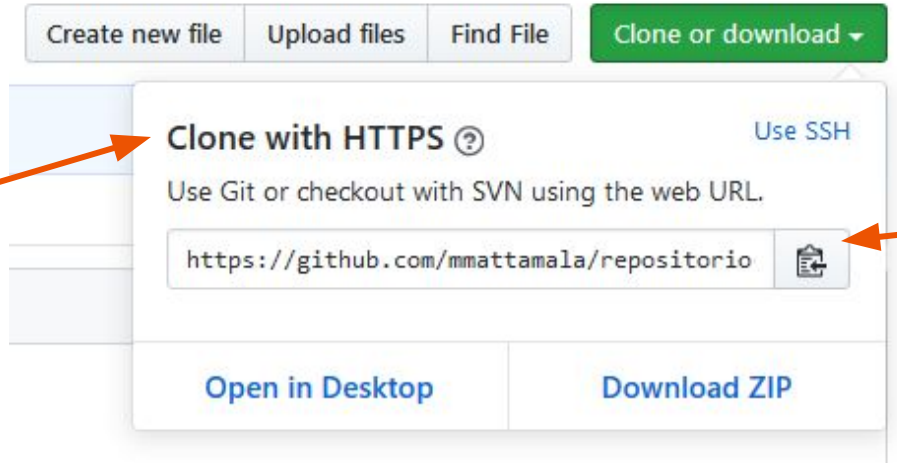
Paso 4: Revisar el repositorio creado



Archivo **README**

Contenidos del **README**

Paso 5: Obtener la dirección para clonar el repo



Asegurar que la opción con **HTTPS** esté seleccionada

Copiar la dirección del repositorio

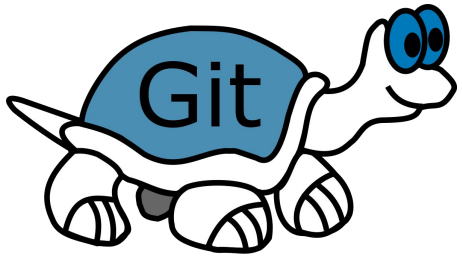
Clonando repositorios en nuestros pcs

Dos formas: Git Bash o Tortoise Git

Git Bash

Viene instalado con Git

Operación por **línea de comandos** (similar a Linux y MacOS)



Tortoise Git

Requiere instalación (tortoisegit.org/)

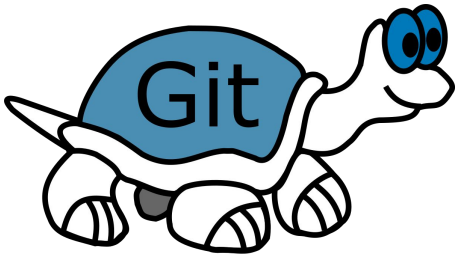
Operación con **interfaz gráfica (GUI)**

Dos formas: Git Bash o Tortoise Git

Git Bash

Viene instalado con Git

Operación por **línea de comandos** (similar a Linux y MacOS)

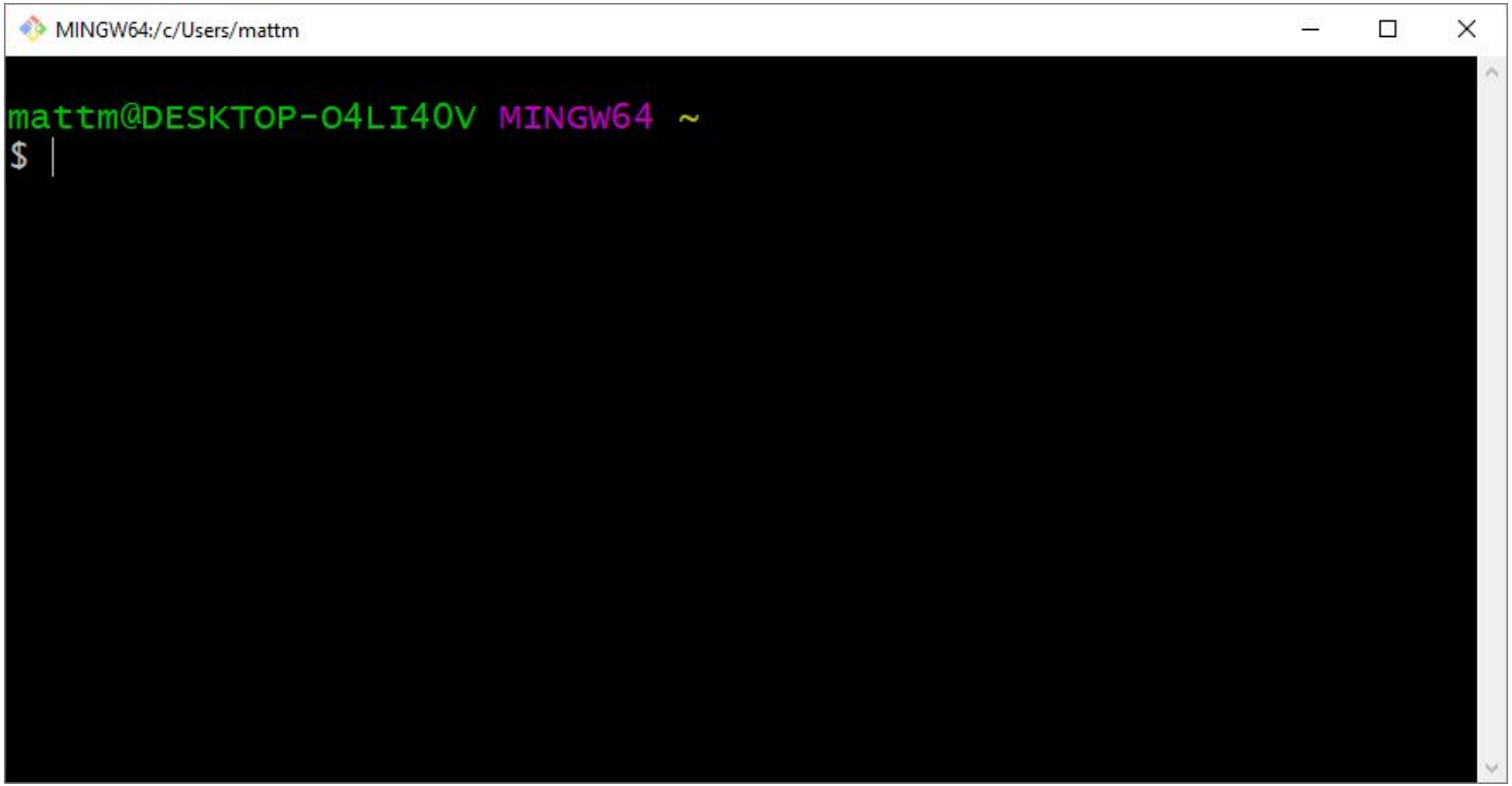


Tortoise Git

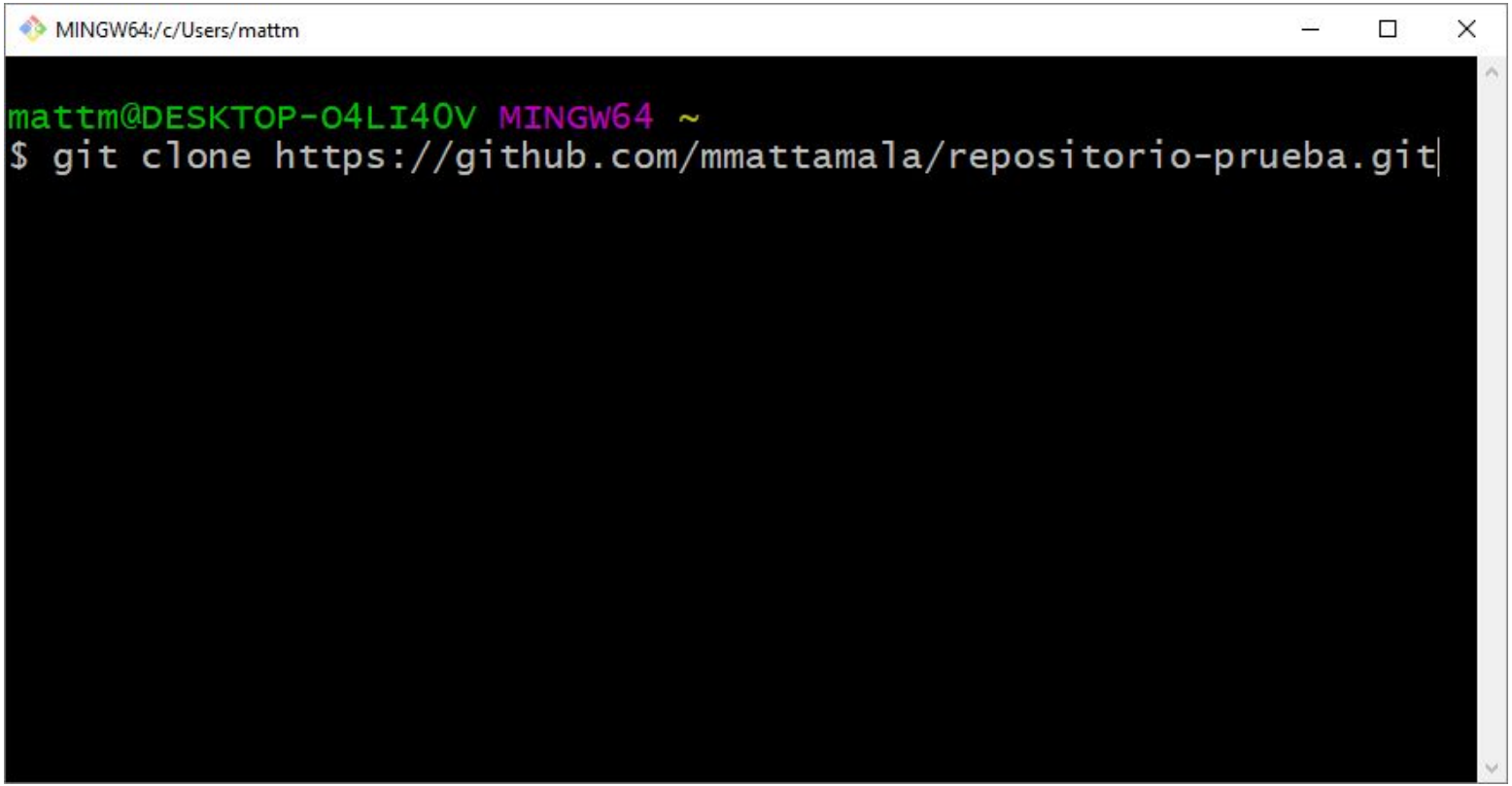
Requiere instalación (tortoisegit.org/)

Operación con **interfaz gráfica (GUI)**

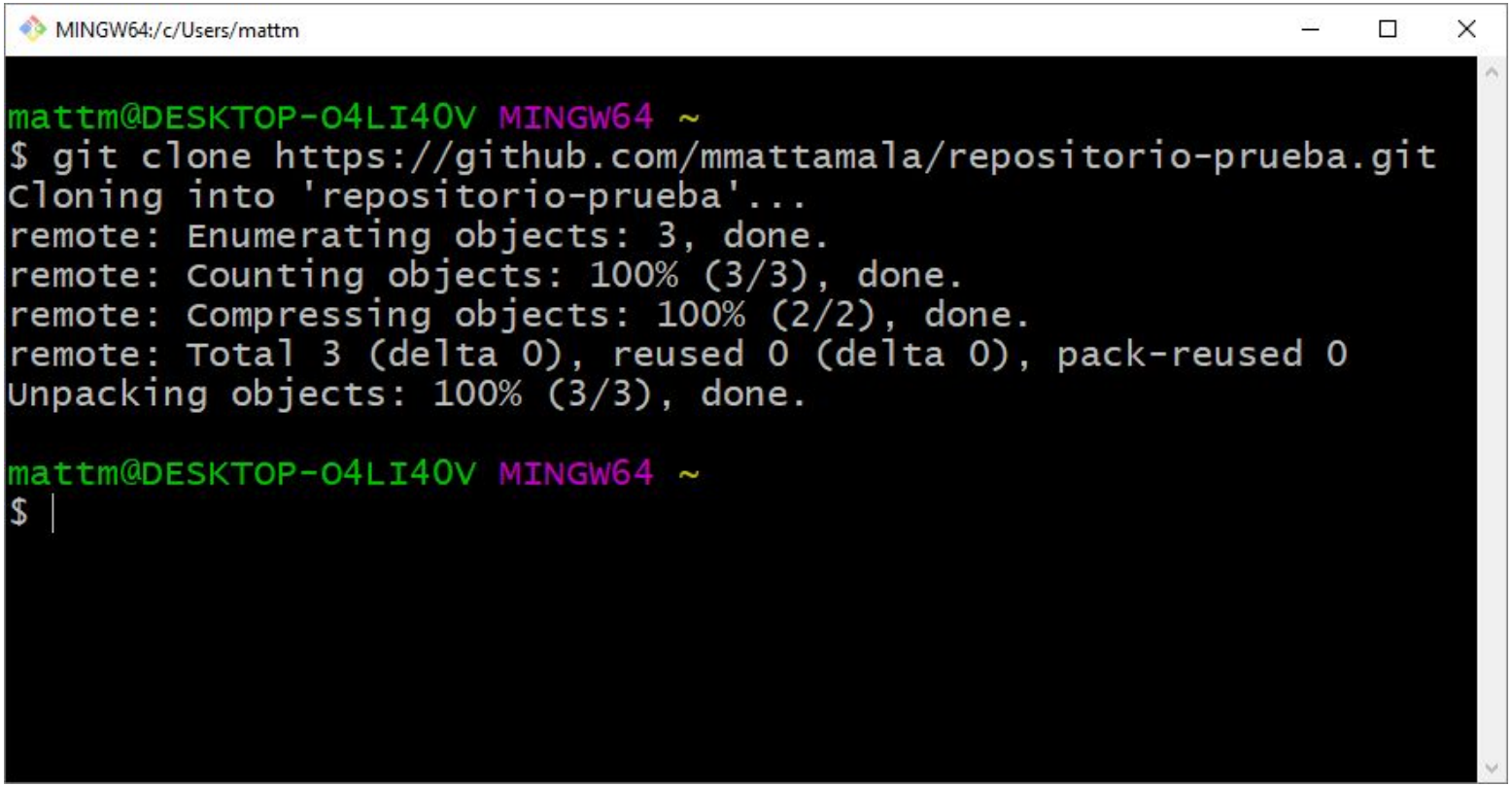
Paso 1: Abrir Git Bash en el pc



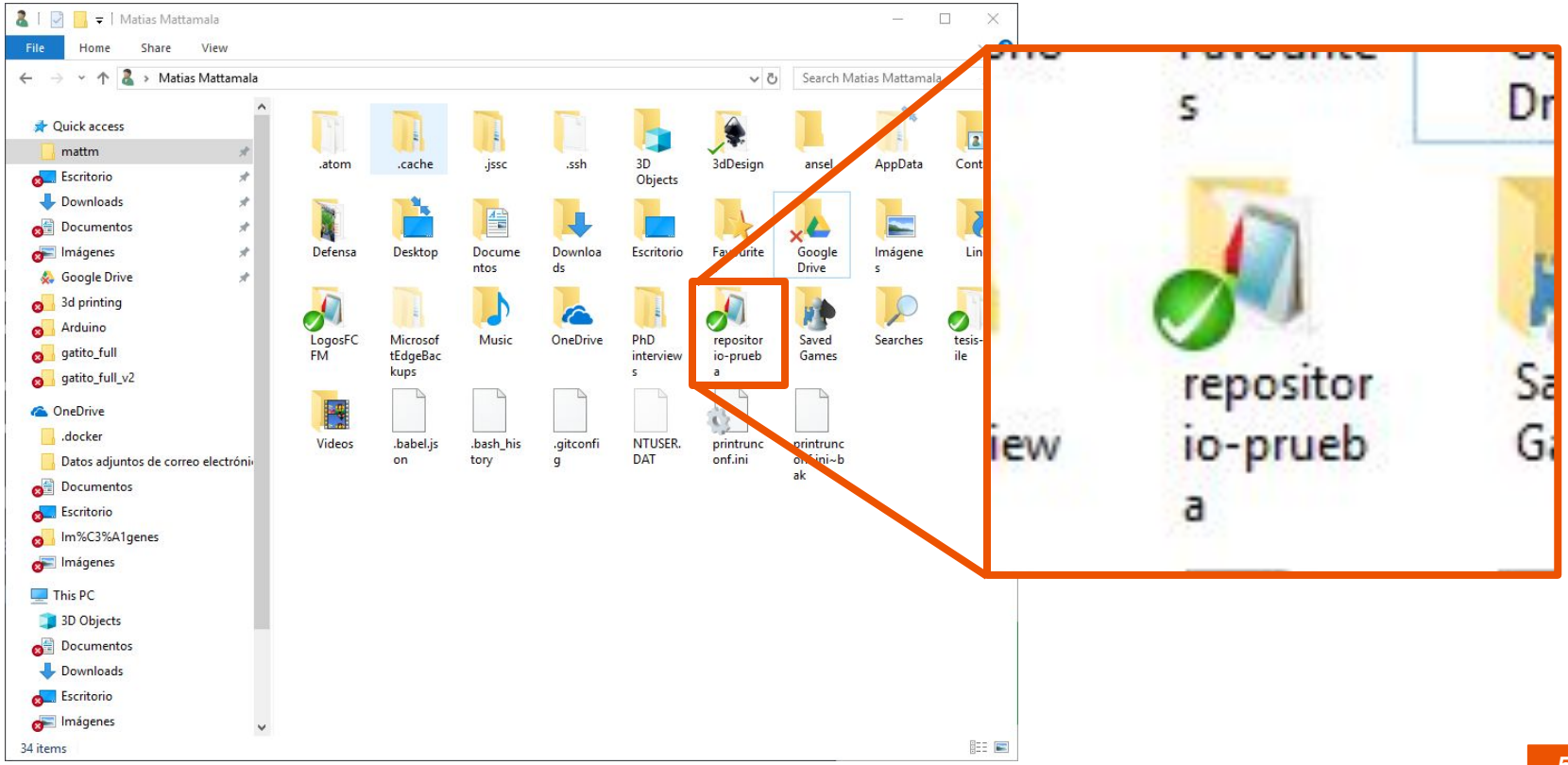
Paso 2: Clonar el repositorio



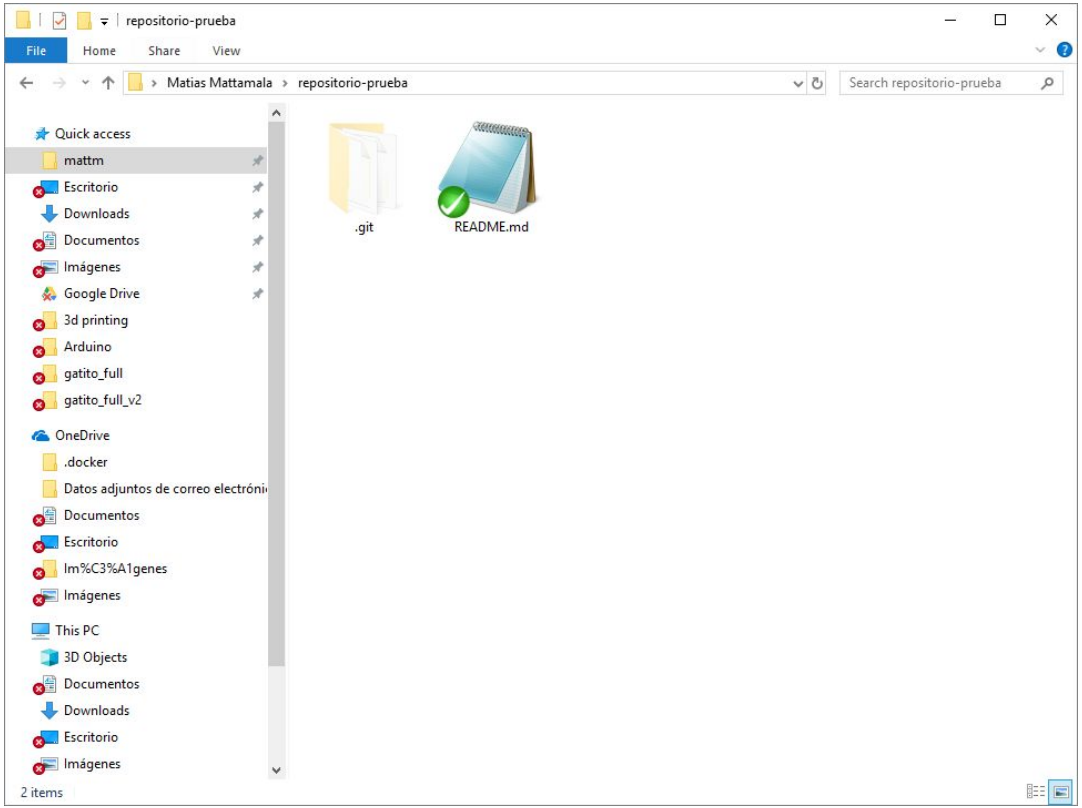
Paso 3: Clonar el repositorio



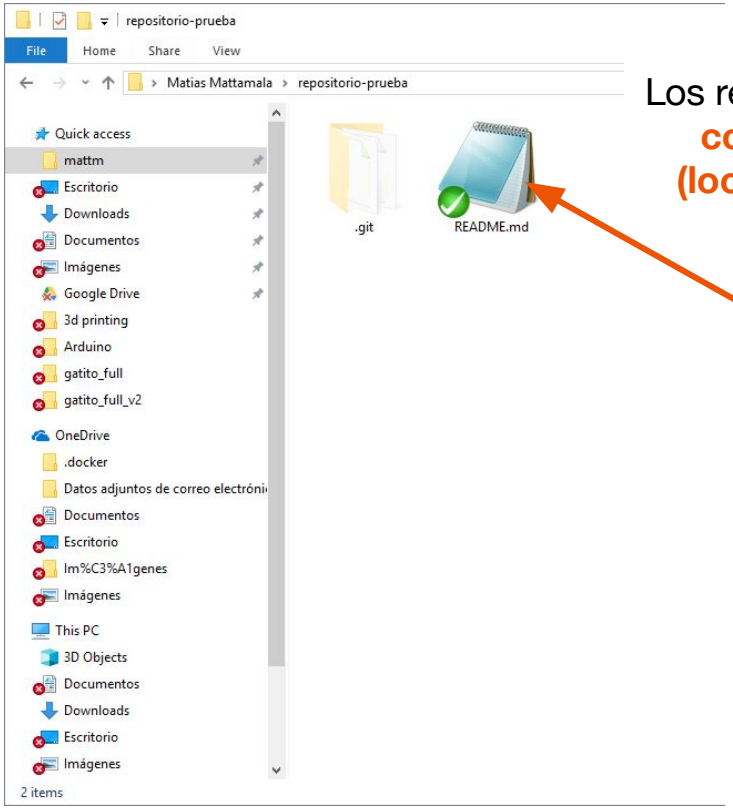
Paso 4: Revisar que el repositorio esté creado



Paso 5: Revisar dentro de la carpeta

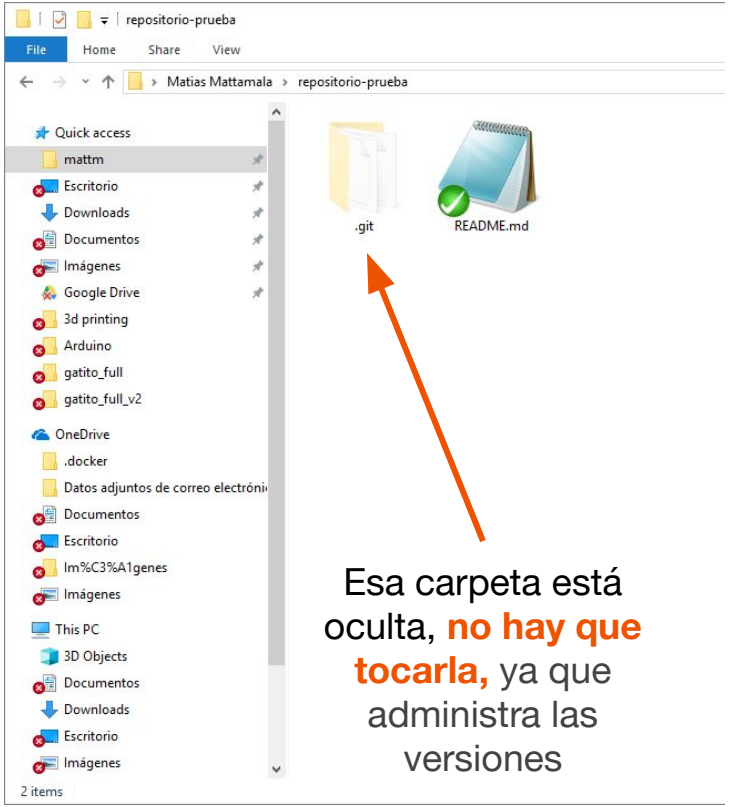


Paso 5: Revisar dentro de la carpeta



Los repositorios son **consistentes (local y remoto)**

Paso 5: Revisar dentro de la carpeta **(Importante)**



Esa carpeta está oculta, **no hay que tocarla,** ya que administra las versiones

Actualizando archivos en el repositorio (local)

Formas de editar un repositorio

1. **Editar un archivo**

Genera cambios en el historial de cambios del archivo

2. **Eliminar un archivo**

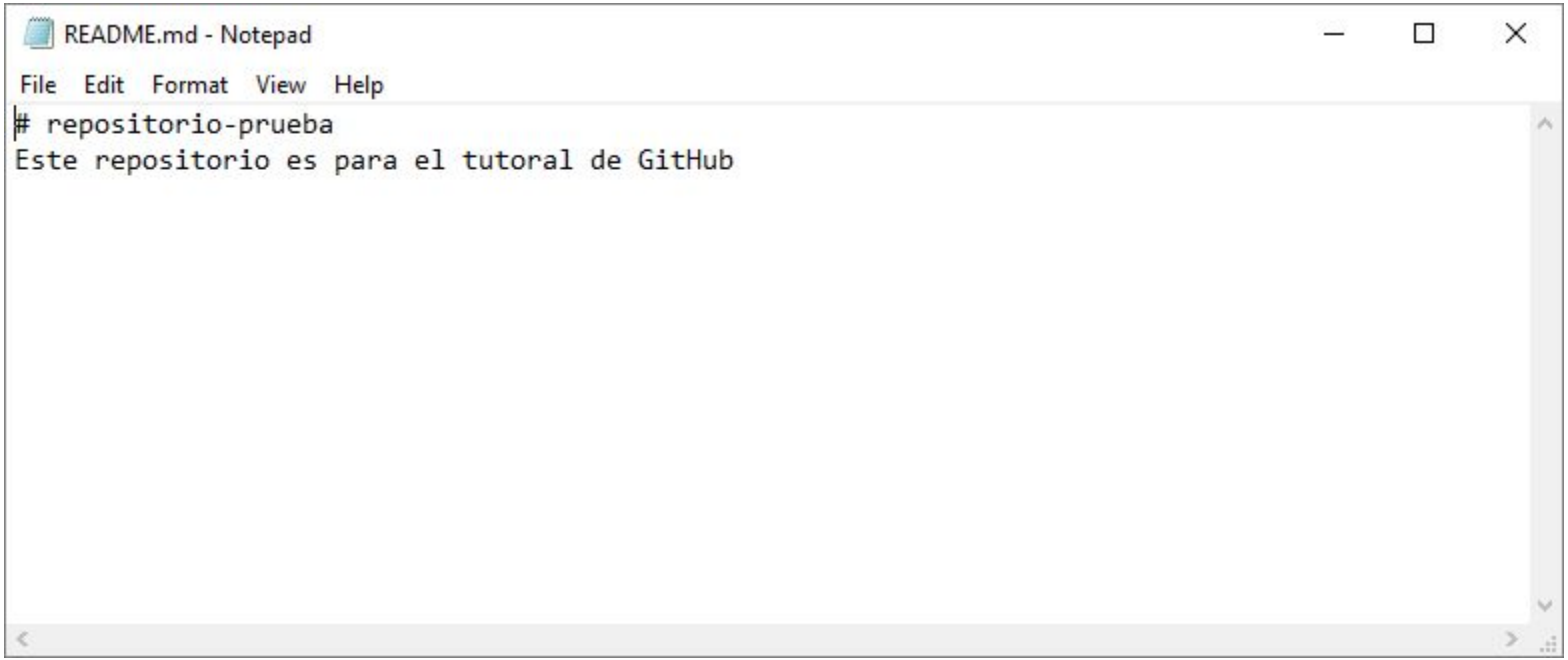
Modifica la estructura del repositorio

3. **Crear un archivo**

Genera nueva información en la estructura del repositorio

Todo cambio sigue una **secuencia de pasos similar** para actualizar el repositorio local y remoto

Caso 1: Modificar archivo (notepad)



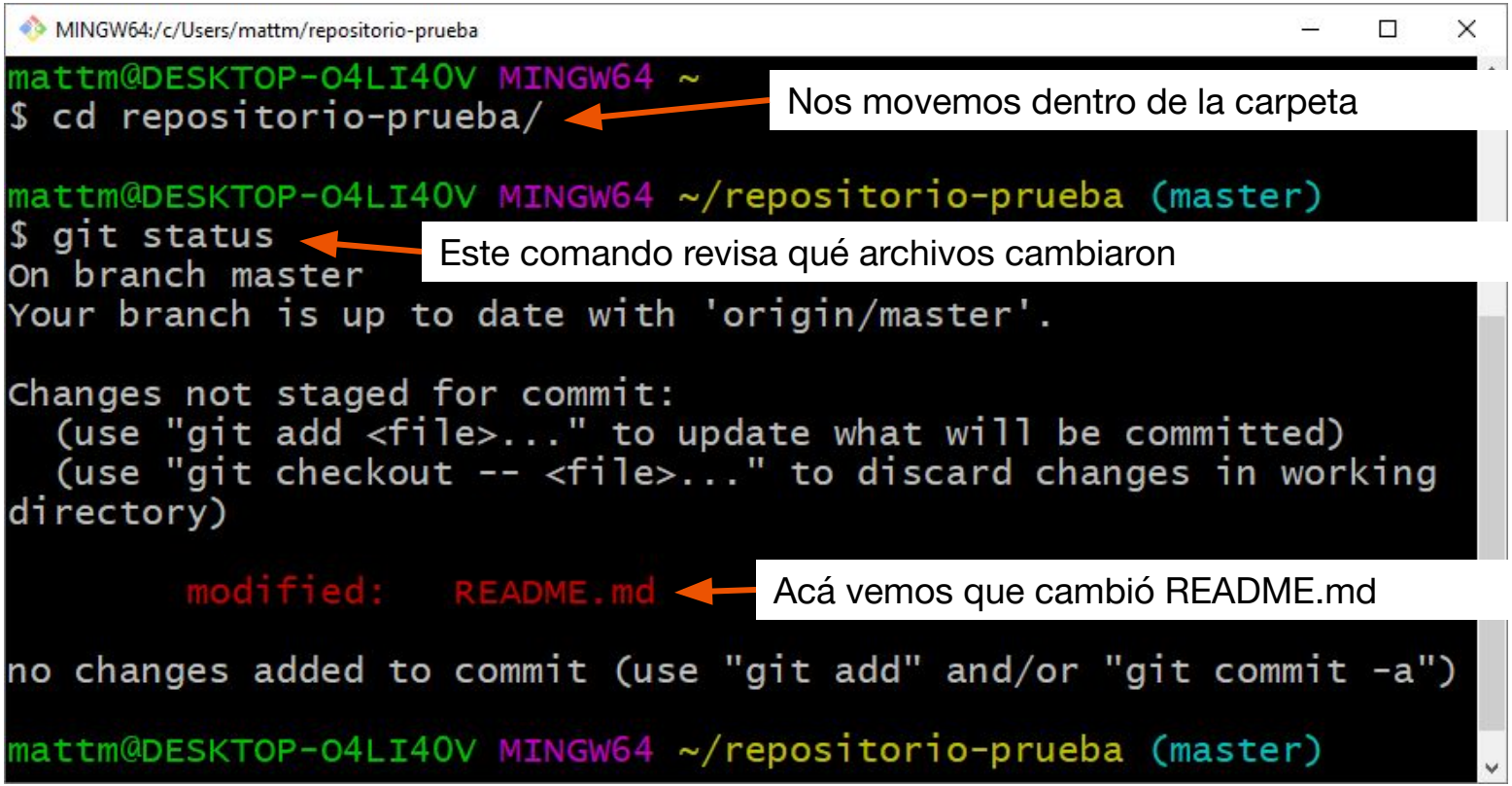
Paso 1: Agregar nuevas líneas

README.md - Notepad

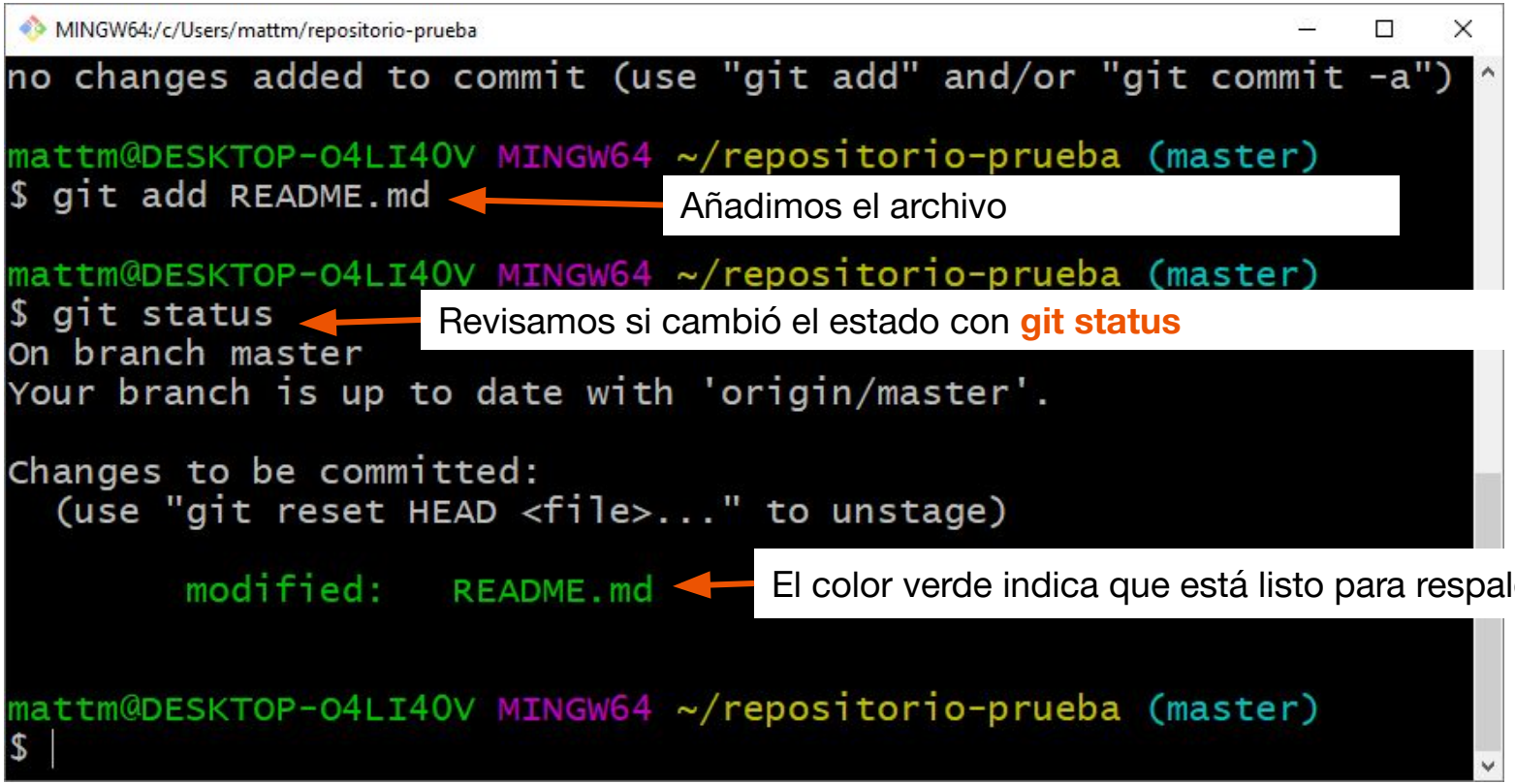
File Edit Format View Help

```
# repositorio-prueba
Este repositorio es para el tutoral de GitHub

## Ahora agregaremos un subtitulo
Para incluir mas informacion

## Y otro subtitulo
Y algunas palabras en **negrita**, _cursiva_ y como `codigo`
```

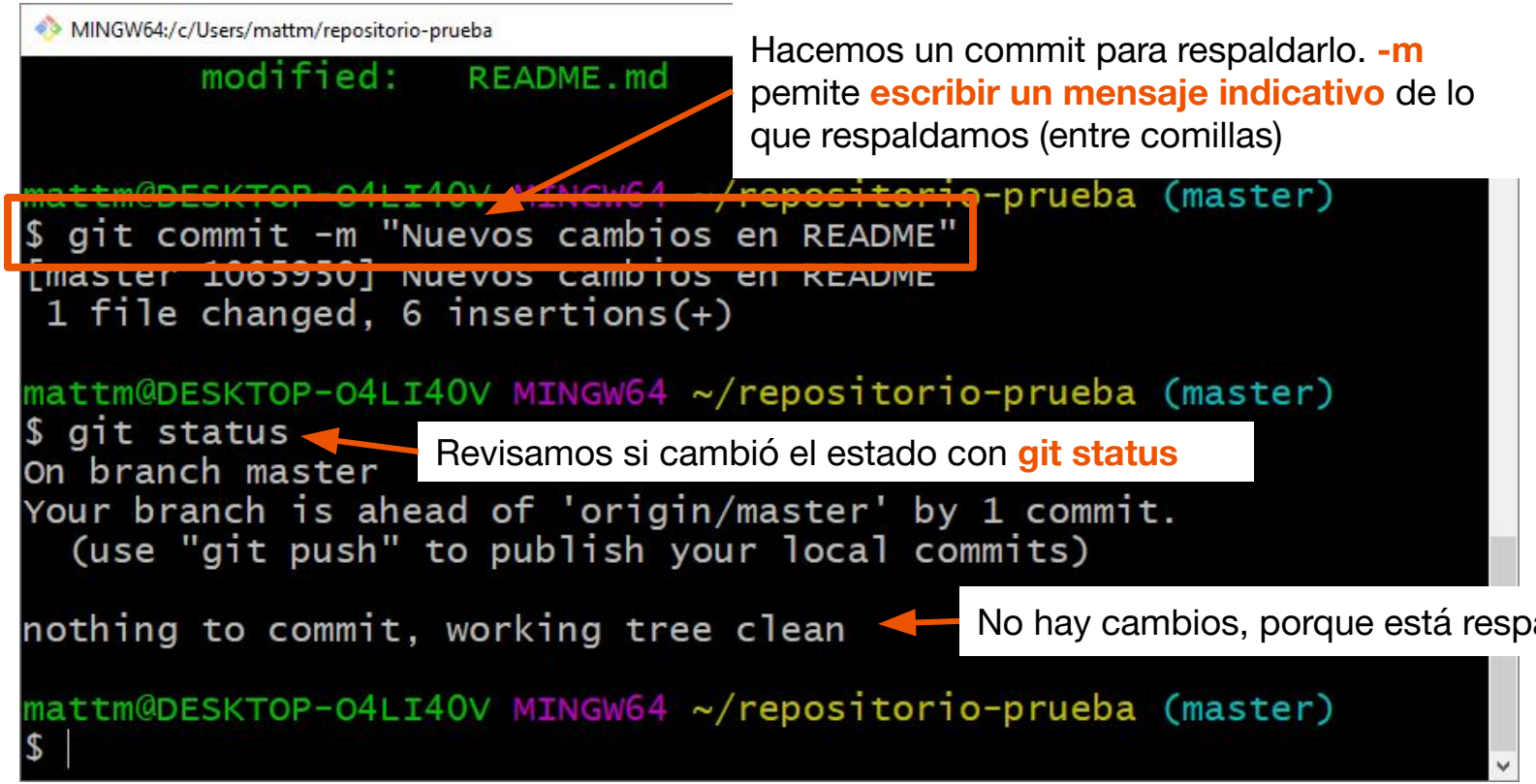
Paso 2: Revisar los cambios usando **git status**



Paso 3: Registrar los archivos con cambios (**git add**)



Paso 4: Crear un respaldo (**git commit**)

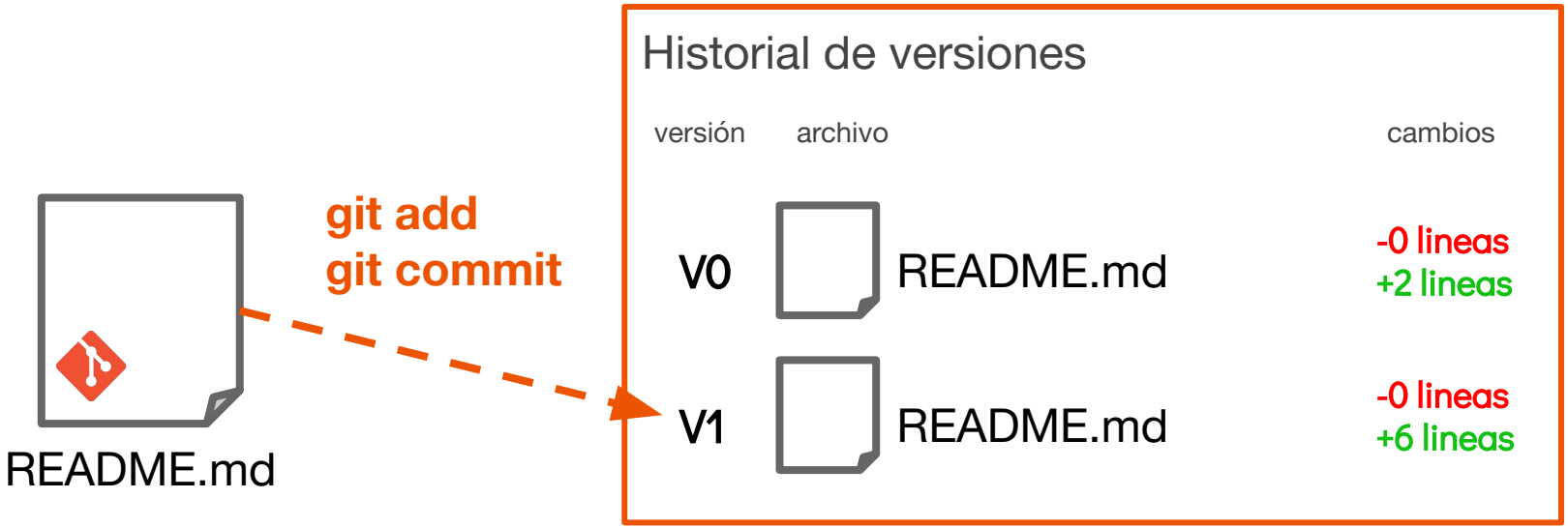


Hacemos un commit para respaldarlo. **-m** pemite **escribir un mensaje indicativo** de lo que respaldamos (entre comillas)

Revisamos si cambió el estado con **git status**

No hay cambios, porque está respaldado

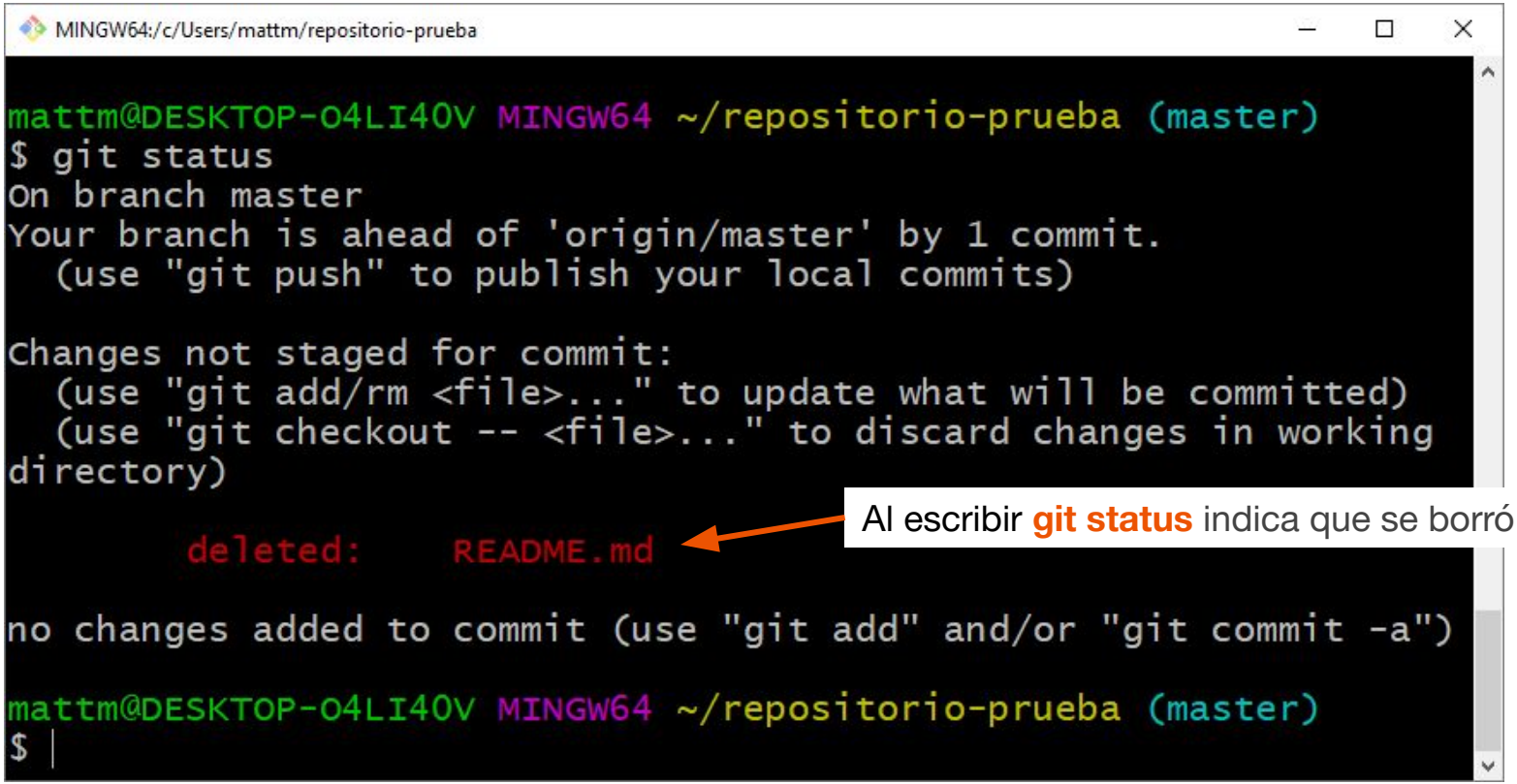
Qué acabamos de hacer



Observación: Para ver el historial se puede usar el comando **git log,** que además muestra los **mensajes de cada commit**

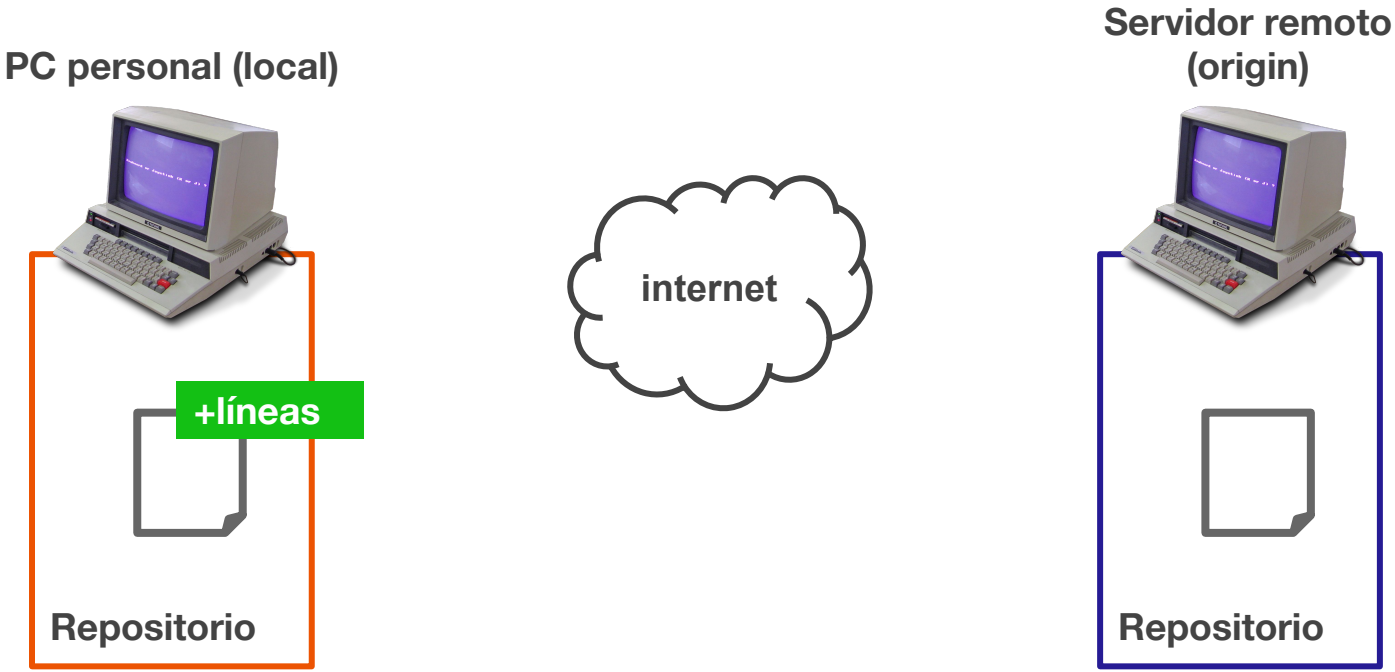
Qué cambia si **creamos** archivos

Qué cambia si **eliminamos** archivos



Al escribir **git status** indica que se borró el archivo

Estos cambios solamente están en nuestros pcs



Actualizando archivos en el repositorio remoto (GitHub)

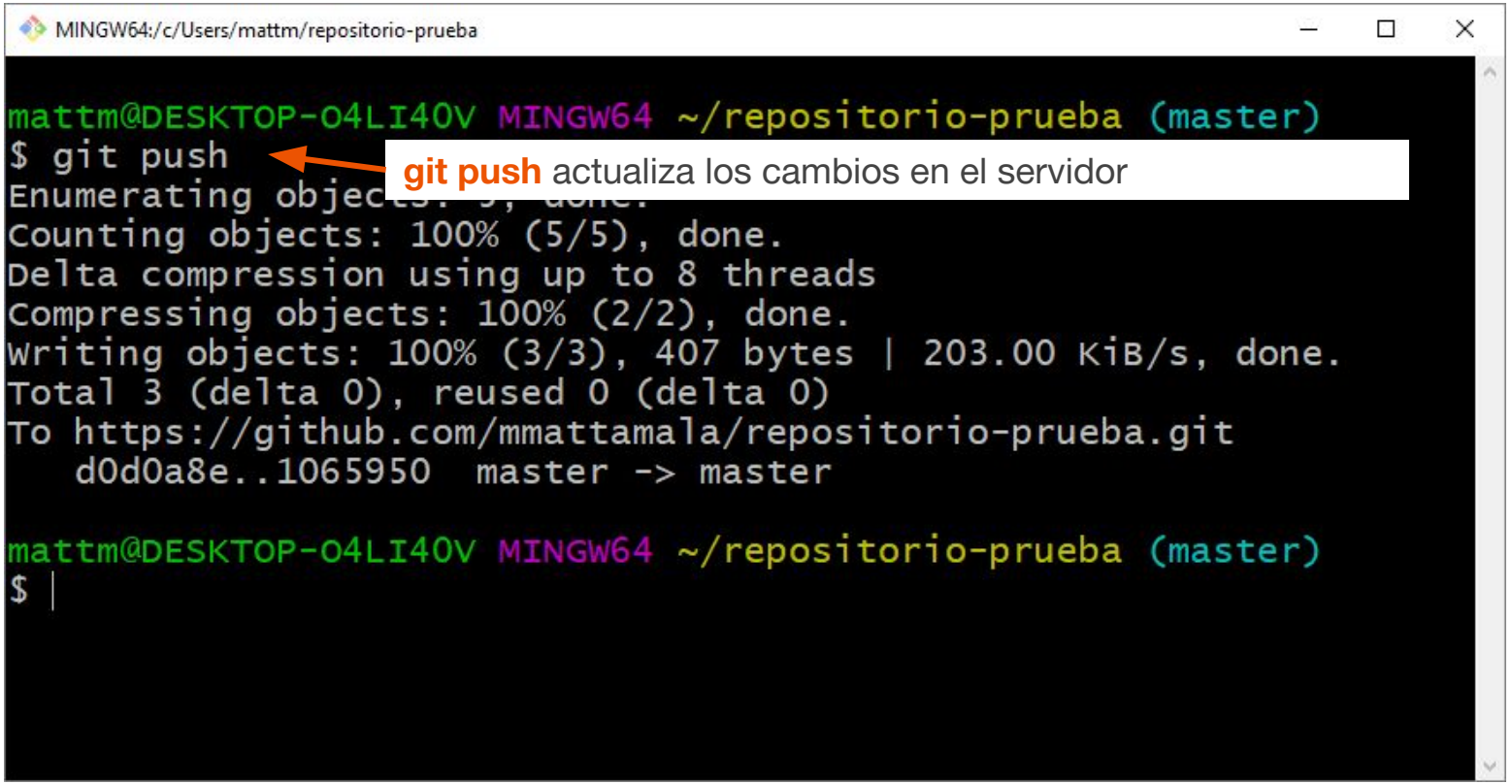
Paso 1: Asegurar que todo esté “**commiteado**”

git status

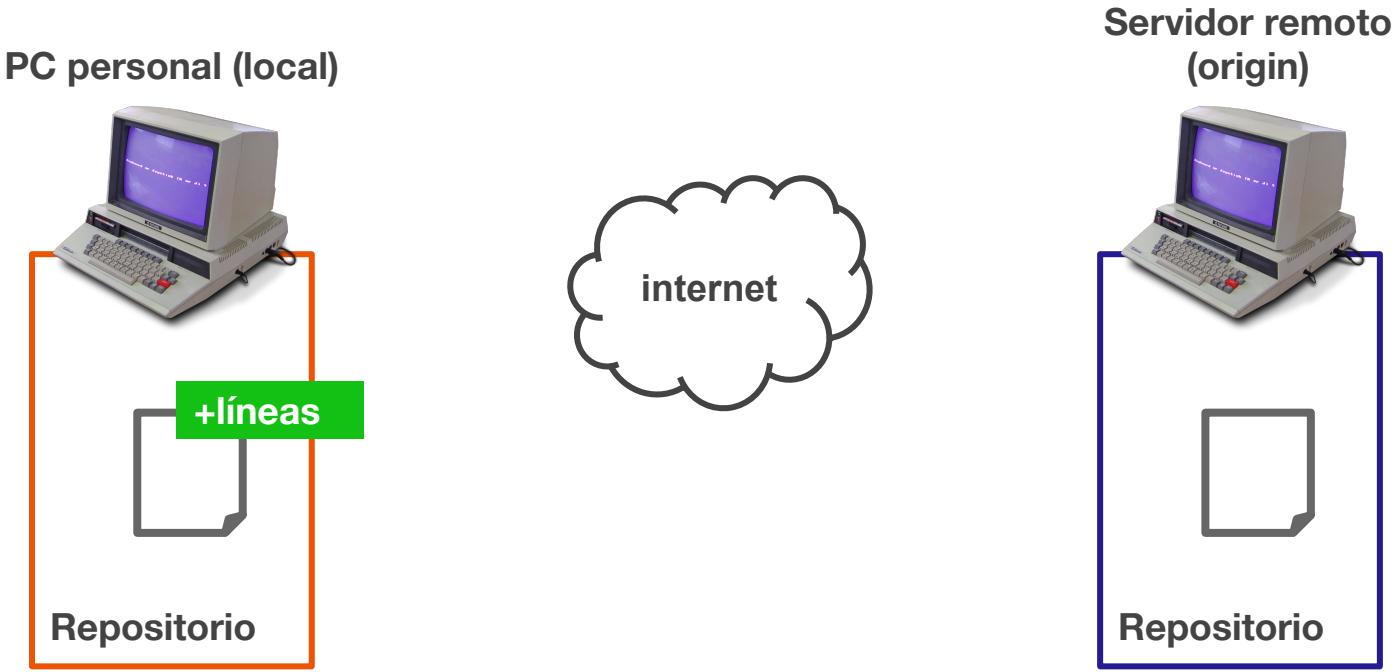
git add archivo1 archivo2 … archivoN

git commit -m *“Mensaje explicativo de los nuevos archivos/cambios”*

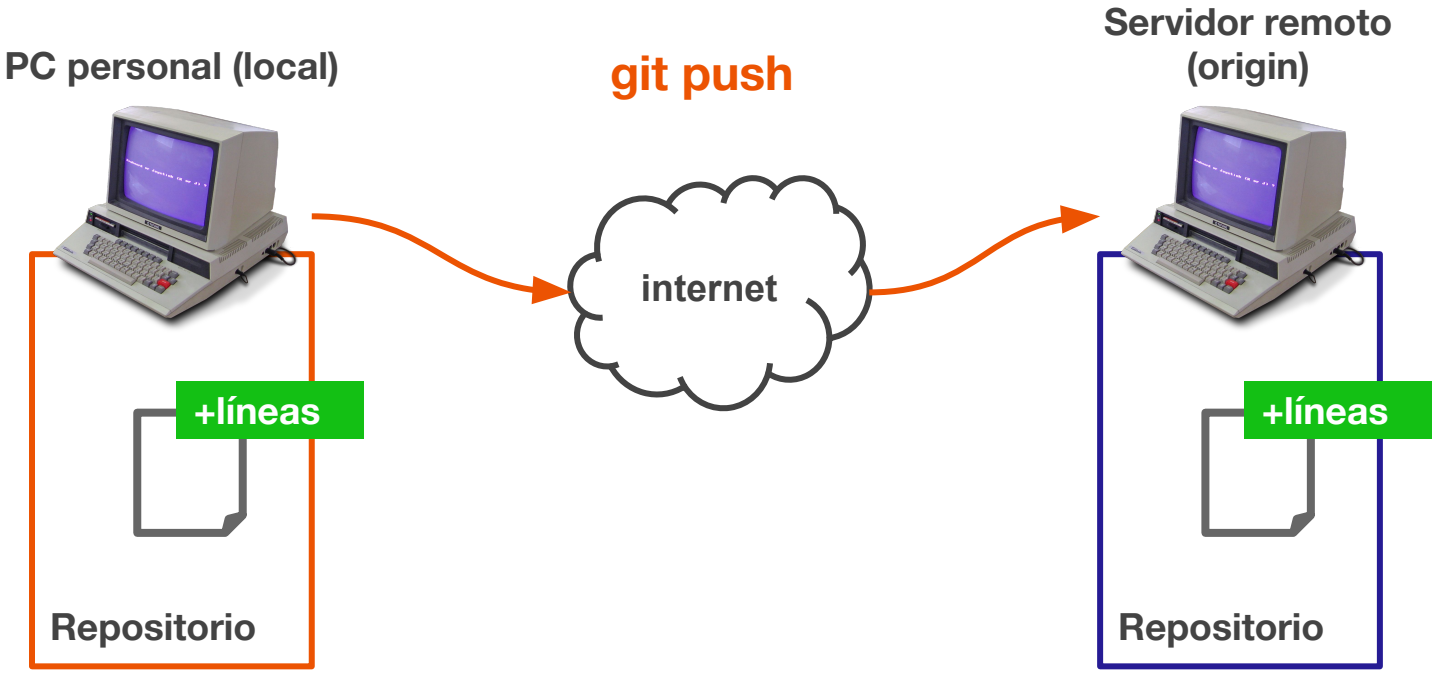
Paso 2: **Pushear** los cambios



Qué acabamos de hacer



Qué acabamos de hacer

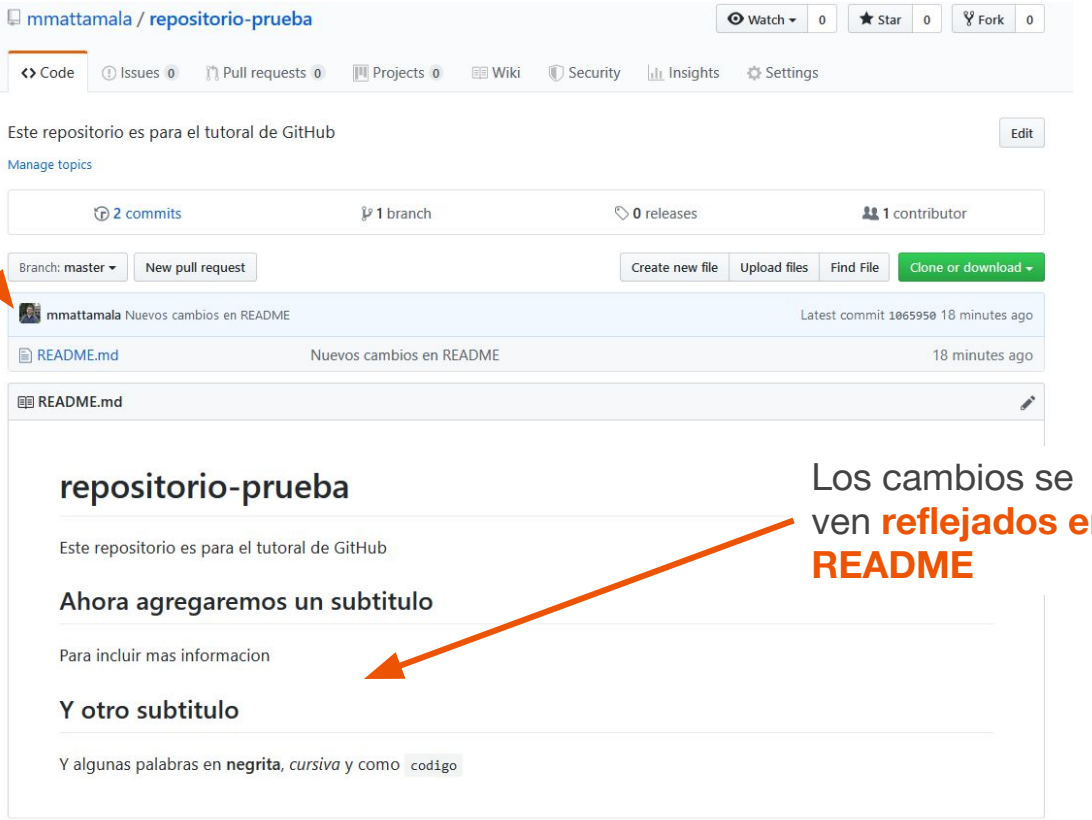


Qué acabamos de hacer

Los cambios están
respaldados en el
computador local y GitHub

¿Cómo confirmarlo?

Aparece el mensaje que escribimos



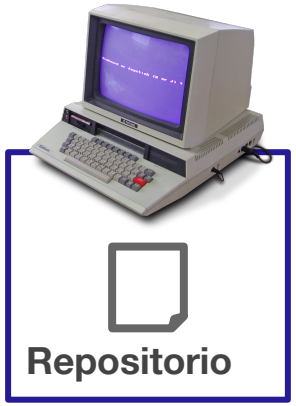
Los cambios se ven **reflejados en README**

Temas avanzados I:
Sincronización entre varios pcs usando GitHub

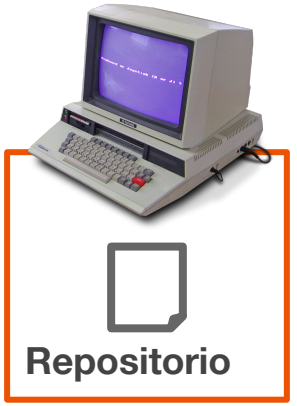
Trabajo con múltiples pcs

PC personal (local)

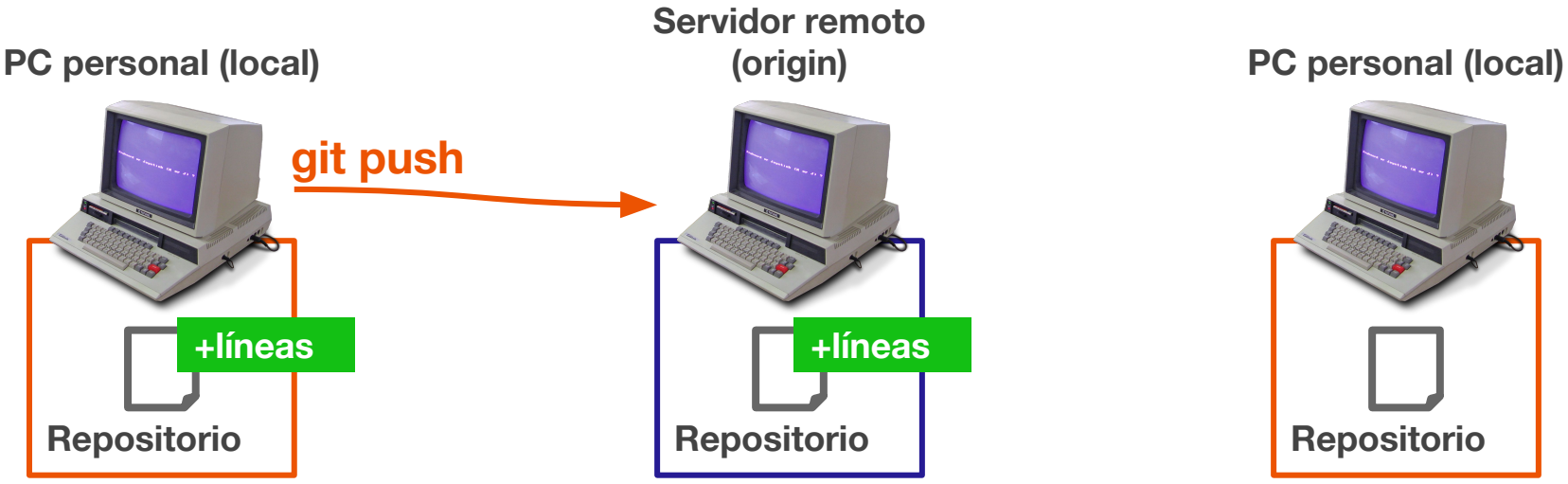
Servidor remoto (origin)



PC personal (local)

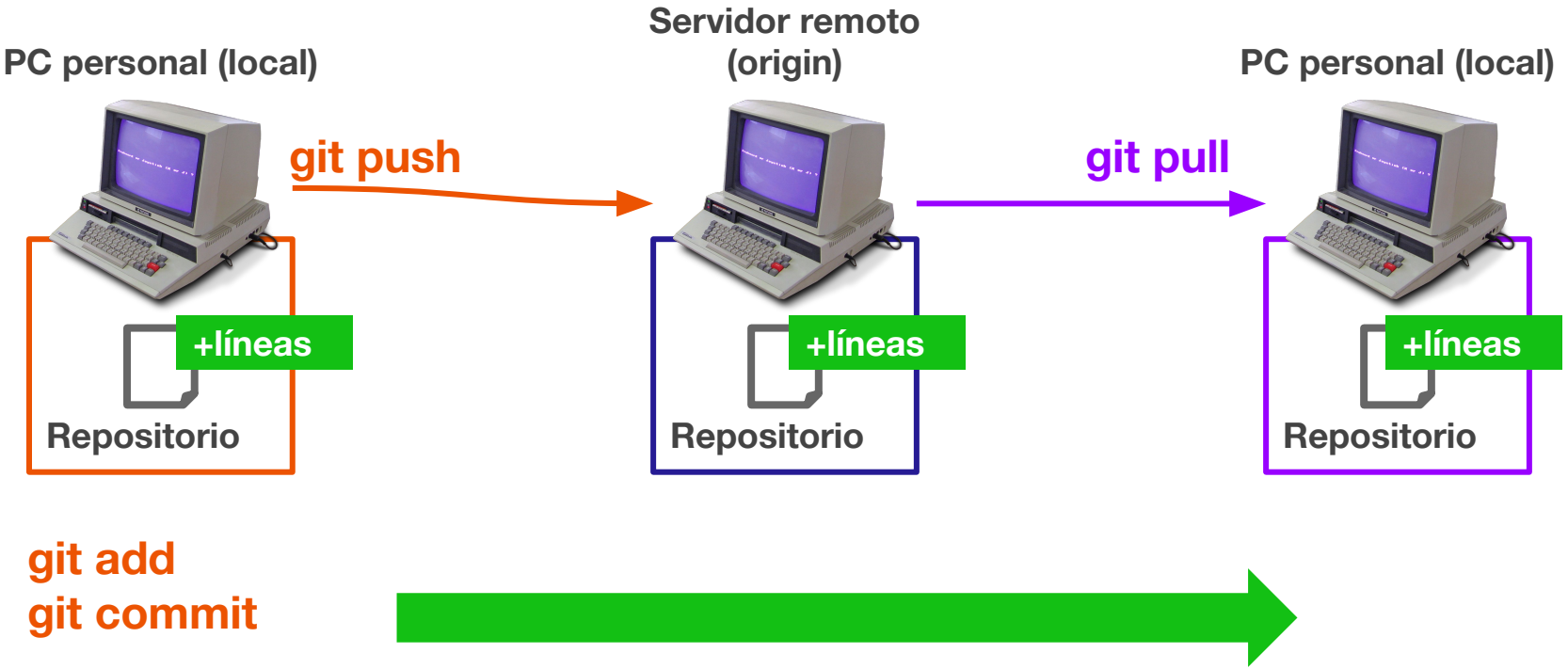


Trabajo con múltiples pcs

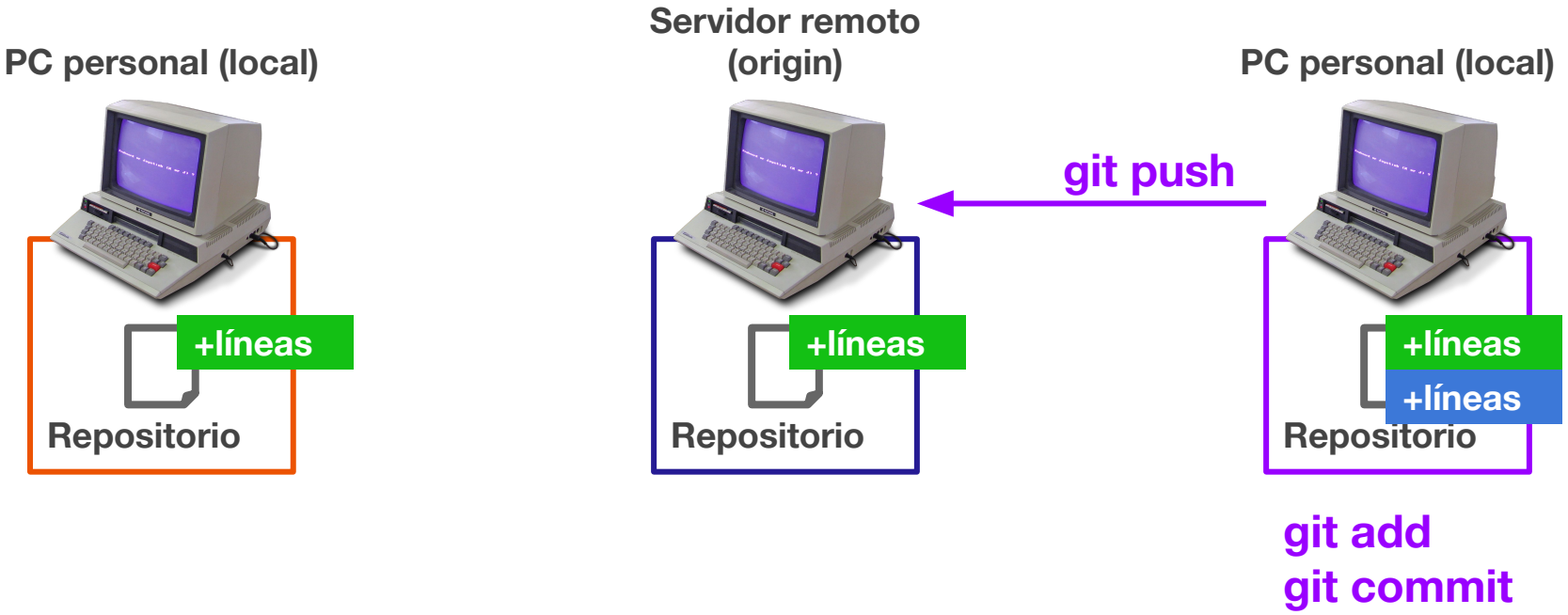


git add
git commit

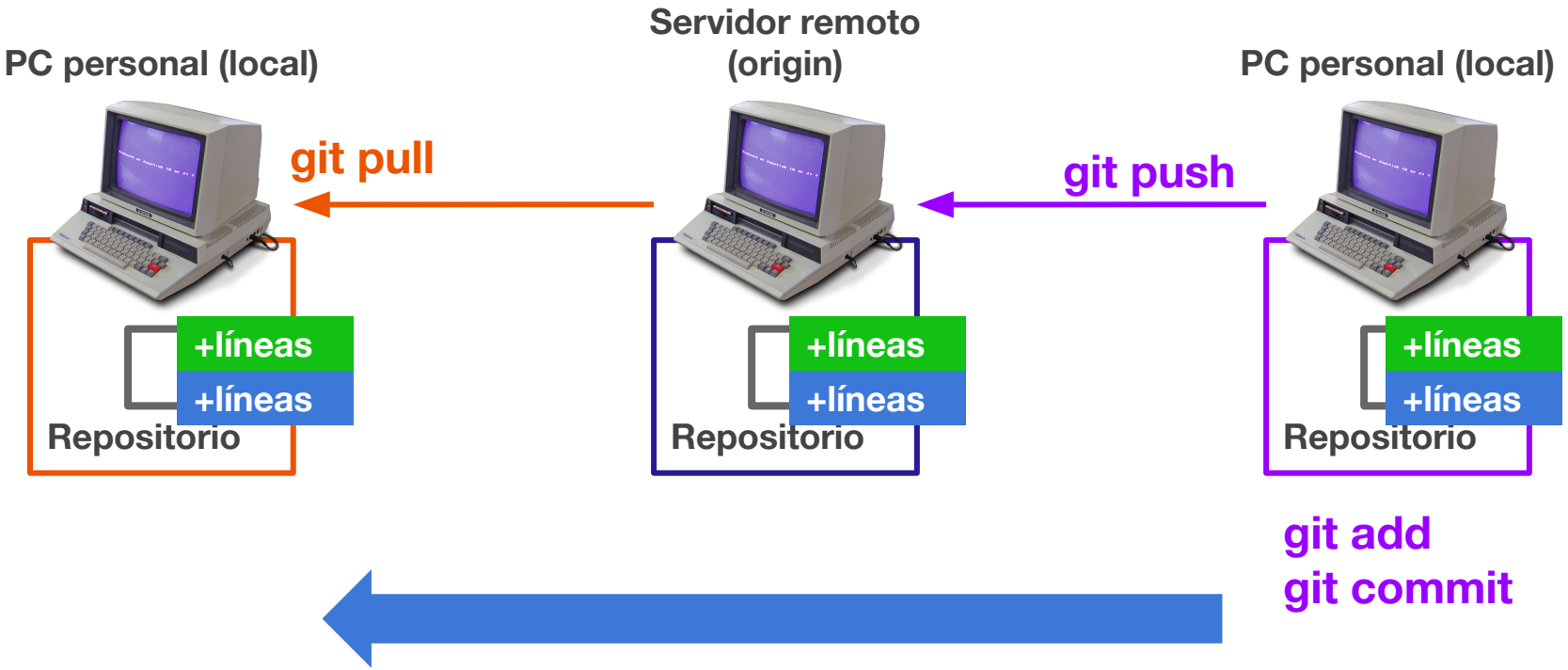
Trabajo con múltiples pcs



Trabajo con múltiples pcs



Trabajo con múltiples pcs



Temas avanzados:
Trabajo con ramas (branches)

Pendiente